DATA INVESTIGATION

EDA the Right Way

Understand and Analyse data
by learning the skill of Assumption
making

Vivek Chaudhary

Anirudh Dayma

John Gabriel

Manvendra Singh

Table of Contents

Acknowledgement

I would love to express my special thanks to John, Anirudh and Manvendra for their continuous dedication to make it a good one. Thanks to Naresh Talwar for supporting me during my hard times.

Special love & thanks to my lovely dog “Kittu” & “Chuggu” for being a part of my unreached destination.

“Kittu” & “Chuggu” will always be missed.

- Vivek Chaudhary

I would like to thank my Mumma, Papa and my Sisters for pushing me whenever I got demotivated. It would have not been possible to complete this book without their support. I would also like to thank the co-authors for making this happen.

And last but not the least, I would like to thank all my friends. Romit and Pushkar you rock!!!

- Anirudh Dayma

I want to thank Vivek Chaudhary for giving me this opportunity to contribute a part of writing this book. I’ve never ever dreamed and thought of writing a book in my life but thanks to Vivek for making this happen. I would also like to extend my thanks to Anirudh, for transferring his experience.

- John Gabriel

First of all I would like to thank god, with whose grace we were able to write and complete this book and also the Co-authors of this book. I am really thankful for the support that I have received from my parents, my Mentor Miss. Sutithi Chakraborty & my friends.

- Manvendra Singh

Special thanks to Sinku Kumar for his time and for helping us when we got stuck. Thanks for your selfless support and efforts.

Gratitude to Swati Mishra for helping us out with Applied statistics code snippets.

We would also like to thank the below people for devoting their precious time for proof reading this book (names in alphabetical order).

Ayesha Khan

Bharadwaj Narayanam

Ritama Daw

Urmi Shah

Also, thanks to everyone who was associated with this book directly or indirectly. We hope we didn’t miss anyone.

Preface

Why This Book?

Normally a human being will watch a movie when it attracts him in the first 30 mins or 20 mins. Everybody will expect something new which covers their interest. If we can make a difference out of all the other movies available in the market, for sure it is going to hit a BLOCKBUSTER. Having that in mind, we have come up with a book named **“Data Investigation - EDA The Right Way”**. It will tell you how to approach a problem statement by making assumptions before you aim for building a Model with 92% accuracy. This book will give you a proper sense of EDA. It will help you understand the exact intuition behind EDA.

Did you ever think, how you will come to know which graph you should plot like histogram, pie chart and which are the different techniques to have a look at your Dataset? Interestingly, **EDA is the process to know which one fits the best** and much more beyond that.

Let's consider you have three different features having 20%, 40%, and 60% Null values, here most of us would delete those features which have Null values more than 40 % or 50%. But, this is not how exactly it is done. There are multiple techniques to fill Null Values but which technique to choose is something we will come to know by having a deep understanding of the problem statement, this is known to be EDA process and will be covered in this book.

For example, let’s say that you delete a feature which has around 60% Null values, but what if that feature is really important while building your Model. By this small mistake of deleting that feature, we will lose some important information and this is going to impact our model. That's why the **research, patience, and critical thinking with commonsense** concerning a problem statement will help us tackle this situation which is also a part of EDA.

Let's understand in layman terms, what we want to deliver to our audience. For example, you want to cook Rajma and you get the recipe from your mom. Ingredients include garam masala, ginger, salt, ghee, and oil. But what would happen to the taste of Rajma if you don't know how much quantity of ingredients you want to add for a good taste? So basically, we all know that cooking rajma is a technique that we have learned from our mom but what quantity of ingredients to add and when to add before we cook Rajma, is something that should be known. And this book is going to deliver the questions of what, when, and how to add ingredients for a good taste of Rajma(Before building a model).

The reason behind writing this book:

There are many reasons. But to keep it simple, this book is not going to start defining **"What is Data Cleaning? How to present the same in the visuals"?** But rather, we are going to follow a unique way, where we'll handle EDA domain wise, step by step in a simplified way. Because the core idea behind EDA what we believe is,

"EDA is a process of approaching the problem statement before Model building."

Individuals mostly focus on visualizations like stacked bar charts, 3D charts, etc, they majorly concentrate on making fancy graphs without actually getting the motive of that visualization. You should have some set of questions in mind before you make a visualization. If you cannot explain **"Why have you made a particular visualization?"** then trust me, it's of no use. So EDA is much more beyond those fancy visualizations and that is something which would be covered in this book.

Yourself give me an answer, which visualization do you think would be best when you tackle a usecase? Most of us would not have an answer for the same. We have to understand that, plotting any graph without knowing the reason behind what exactly we are looking for from such plots will not land us to a right conclusion, until and unless we are clear with our Why's. So, we may be thinking where we get an answer to our **"Why's".**

The only way is to have a deep understanding of the problem statement, make assumptions out of it and then proceed to plot the graph. But again, we have to find insights from that particular plot i.e. check if our assumptions are correct, if not then what other insights we can draw from that graph and then make some more assumptions to proceed with.

Remember **'Assumption making is the key skill'**, and even wrong assumptions may lead you to the right fit.

How is this book different?

This book is written with the motive to unfold some of the unseen aspects of EDA. It's not that we won't be using visualizations in this book but they would be used for a purpose and we would have some set of questions that we would answer with the help of the visualizations. This book will show **that EDA isn't only about replacing NANs with mean/median/mode**; it's way beyond that.

Most of us would initially focus on building an ML model, but in the end, that is going to be the wrong approach if we are not understanding the problem statement. Let's say we have to build a Machine Learning model to detect a *' Bot or not a Bot'*, here we will be focussed to building a model that can detect when it is a 'Bot', but if our model fails to detect when it is 'not a Bot' then our model is memorizing and not learning.

Did you ever think whenever you deal with any data set, why is Machine Learning needed? and if it's needed, then at what different instances your model can be the best and also fail at some point in time?

In this book, our focus into to build models rather we would be focussing on some of the techniques you should follow for EDA when you deal with any problem statement.

This book contains every basic approach that as a Data Scientist we have to deal with. Majority of us have questions about how to apply feature engineering and we will get the answer to the same in this book by working on some usecases from different domains.

For whom this book belongs to?

The book ***"Data Investigation-EDA the Right Way"*** is going to help all the beginners and intermediates who are struggling and finding EDA as a huge challenge. This book is for the Data Science Enthusiast who have just jumped into this field and are busy plotting graphs and filling NANs with mean/median/mode. This book is also for people who have undergone learning but lack when it comes to answering "Why this visualization?" At the same time, this book is also for people who think EDA is just about flaunting fancy visualizations. So this book is for everyone who has a basic knowledge of Python/ML and thinks he/she knows EDA as this book will help them to understand and apply EDA the right way.

One main misconception that people have is that Machine Learning is all about building models. But people fail to understand that model might help to a certain extent but after that, it is the features provided to the model that helps. Machine learning works on the principle of **"Garbage In Garbage Out",** so if we feed poor features to the model, the performance of the model will also be poor.

So to build efficient models one should majorly focus on the features and use EDA to analyse the features and then finally come up with important features that should then be fed to the model. In this book, we will describe how to make assumptions, validate those assumptions, and then come up with useful features to build a model.

Road map

Let's discuss what you are going to learn from this book and what it contains for the readers.

As we all know that Data Science is booming, but we believe that, **'Data Science is not Everyone's Cup of Tea**.'

People tend to follow the wrong practice by focusing on building Machine Learning Model with 92% or 95% accuracy without realizing that their model is learning or memorizing.

Suppose that you are teaching your kid how to identify a car, you would describe it by saying that a car has four wheels. Whenever that kid sees a four-wheeled object it would identify it a car, but do you remember the Mr. Bean show, there was a blue-colored car with 3 wheels.

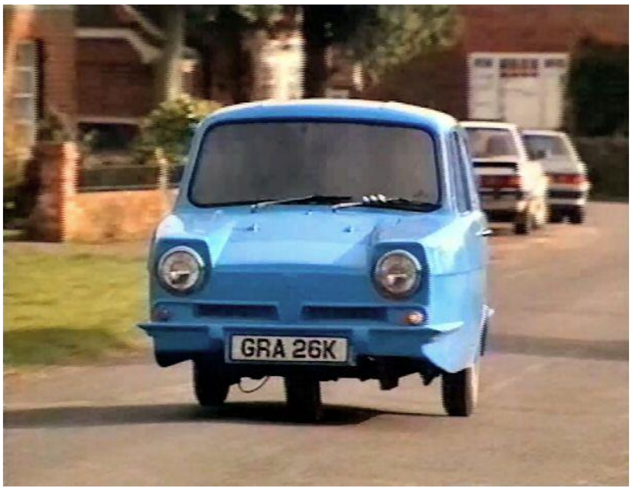


Figure 1: Car[1]

If the kid has memorized then he/she will fail to identify it as a car and the beach bikes with 4 wheels would be identified as a car. In this scenario, we could say that the kid has memorized as he/she gets confused when he/she sees some variation. But if the kid would have learned then he/she would have identified the objects correctly. How the kid learns depends on how well we teach the kid.

[1] Image source - https://i.stack.imgur.com/OuWxL.jpg

Similar is the case with Model building, the model is analogous to the kid and it depends on us how we train the model.

For example, most people blindly apply one-hot encoding or label encoding when dealing with categorical features to build their Model hoping to get good accuracy, they don't bother to think if that encoding technique is going to help or not. The focus should be to make the model learn and not memorize.

Let's say you are trying to teach the same kid to identify an apple and if you just focus on the red color, again the kid would fail when it comes across apples that are green or have a slightly different shade of red. The kid might identify red cherries as apples because the kid knows that whatever is red should be an apple. While making the kid understand, we should also focus on the size and some other variants of apple which have different color/characteristics. So size along with other characteristics becomes an important feature when dealing with the above problem. The motive of this example was to make you understand the importance of feature selection. You should be in a position to identify which feature is important and which isn't.

Most of the Data Science Enthusiasts are scared to do some research and don't bother to understand the intuition behind the encoding techniques and fail to understand its application. For example, if you know about one-hot encoding and label encoding then you are going to apply these two every time, which is a wrong practice. Just because you fear to research and experiment, you won't explore which are the other techniques which you could use. By doing some research you might come across some new techniques and it might happen that those techniques won't help you but by doing so, you have learned about some different encoding techniques and also learned about their advantages, disadvantages, where you could apply them and where you can't.

This field is all about exploring and your willingness to make mistakes and learn from those mistakes.

Did you get that?

No?

Let's make it more clear with another example, suppose that you have a Maths exam after 2 days and you are running out of time so you won't be covering all the topics. You would only cover those topics which you feel are important from an exam perspective and you would only focus on those topics. What if things don't happen the way as you want them to happen and what if all the questions come from the topics you have skipped. You would score poor or might even fail.

You should not only focus on the good scenarios but also try to focus on the bad scenarios as well.

And this is how most of the Data Science Enthusiast work, they just focus on good scenarios.

Imagine that you get a chance to work on a project where you have to detect if its a bot or its not a bot. Most of the people would be focussed on building a model that detects a bot and this would be comparatively easy but your model would fail to detect when it isn't a bot and this is a difficult task. Because learning is being able to identify when it's a bot and when it isn't.

What comes to your mind when you hear the term Exploratory Data Analysis (EDA)?

Exploratory Data Analysis isn't only about plotting graphs but it's way beyond that. As the name 'Exploratory' suggests it is more about exploring the data and making yourself familiar with the data, it is about asking questions to the data, making assumptions, and validating them using statistical tests.

For example, if you plan to go on a vacation to Shimla with your family then you are not going to book the tickets directly. Instead, you would look for different places you could visit as per your family discussion, then will select some places based on your budget, etc, and this is something known to be as EDA in layman words.

If you explore the place you plan to visit before your actually visit that place, then why don't you do the same with your problem statement and data in hand. Remember EDA isn't about who plots more number of graphs, it is about who has understood the data well, it is more about drawing insights from the data.

Data Science always revolves around **Why's**. Before plotting a graph you should be clear with 'What are you looking for in the data', you should have some question in hand, some assumptions made, and then check if the data looks the way you assume it to be. Your assumptions might be wrong but they would lead you to the correct conclusion. The graph which you plot is an answer to the questions you have.

Applying common-sense and making Assumptions is one of the key skills you should try to master as a Data Scientist.

It's pretty simple, **Research helps you improve your common sense and Common-sense helps you create assumptions**.

But remember ***Assumptions are always wrong until they have statistical significance*****.** And this is where statistical tests help us and that's the reason we have given a major emphasis on Statistical tests in this book and have it as the very first chapter because to validate your assumptions you need statistical tests.

For example, if someone told us that Virat Kohli's strike rate is similar to Rohit Sharma's strike rate then being a Data Scientist we won't agree upon that statement unless we look into data and check for its statistical significance.

In this scenario,

Null hypothesis - The strike rate of both the batsmen is similar.

Alternate hypothesis - The strike rate of both the batsmen is different.

Before getting into coding, do some research about the problem statement, make some assumptions and even the wrong assumptions might take you to the correct solution.

Intuition behind EDA?

Let us try to understand the intuition behind EDA with the help of an example.

Suppose that you want to cook chicken then you should know which all ingredients are needed to make a good recipe. Some of the common ingredients would be chicken, salt, garam masala, garlic paste etc. Now knowing when to add which ingredient and what quantity to add is an art and it comes only with experience. Same goes with EDA as well, you should be in a position where you know which technique should be used and which should not be used. This will come with practice and you should be willing to make mistakes as mistake is the best teacher.

Suppose the chef forgets to add salt to all the dishes that he cooks, then it won’t taste good, Right?

Similarly if you plan to use Linear Regression (Ordinary Least Square), and if you don’t bother to check if the required assumptions are adhered by the data, then the things won’t work as you wish them to work. Suppose a model assumes that the data should be normally distributed, but the data you have is skewed, in such a case you should apply techniques to handle skewness. Just knowing the techniques isn’t going to help if you don’t use them in real life.

EDA is a process to know when & where to apply a particular technique by doing research.

There are two ways in which a person would send a connection request on Linkedin -

To increase your number of connections, you randomly send a connection request to 'N' number of people & it's somehow same like building an ML model without understanding the problem statement, by doing this you would build a model but it won't be worth it. Same would happen on Linkedin as well, you would increase the number of connections but it won't be worth it.

How would a person like us would send a connection request on Linkedin:

1. At first we will be clear with our interests and would look for people with similar interests.
2. We will go through the Linkedin profile of the person we intend to build a connection with, we will also look at his past experience and projects.
3. Finally we will check if that particular person can add any value to us in terms of learning new things or bringing us across new opportunities. If that's the case, then we would send a connection request.

So, what’s the final intuition that you can make from the above example?

Linkedin is not Facebook, agreed with the intuition?

By now, it would be clear that EDA is about exploring the data, and to do that there are different techniques, one of which is Visualizations and the other is making assumptions, there might be many other techniques as well. But our ultimate goal is to be in a position where we feel like a boss and say ask me anything about it, we should feel as if we own this data.

Figure 2: Feeling Like a Boss[1]

The main motive of this book is to make you aware of some of the weapons which you could use to torture your data and ultimately start **#FEELINGLIKEABOSS.**

EDA is a philosophy which would help you to understand the process of When and Where a particular technique should be applied.

Chapter 1 – The Game of Statistics

Till now you would have clearly understood the importance of making assumptions and validating them using Statistical Tests. So that is what we would be focussing upon in our first chapter. Here we would be dealing with the Importance of Assumptions, Types of Assumptions, Different Statistical tests and when to use which test with Real-life examples so that it becomes easy for you to relate.

[1] Image source - https://imgflip.com/i/4aouuy

If statistical tests bother you, trust us by the end of the first chapter you would feel pretty comfortable with statistical tests and you would get a clear idea about where to use which test.

Chapter 2 – Kaggle Exploration

Once you are comfortable with statistical tests and aware of their usage, that is the time when you need to get your hands dirty on Kaggle. This chapter revolves around Kaggle Exploration, it will tell you what Kaggle is all about, what are some of the common mistakes people make while exploring Kaggle, how to overcome those mistakes, we will also demonstrate how to deal with a problem statement from the very start. It will involve understanding the problem statement, exploring the data, making assumptions, and a lot more things.

Chapter 3 – Churn Prediction

After having an idea about how to explore Kaggle, you are good to go with another problem statement, this would be regarding Churn Prediction. Here we will explain what is a Customer Churn, Why is it important, What steps should you follow while dealing with this problem statement. We won't be solving this use-case end to end by coding but we will give you an idea about how to approach a problem statement. We will also sight approaches followed by Top companies and some of the popular strategies to deal with this problem. In the end, we would list an important challenge you would face when you deal with this problem statement i.e. imbalance in the dataset.

Chapter 4 – Imbalance Classification

In this chapter, we would tell you why it is important to deal with imbalance, what effect does imbalance have on the metrics, which metrics should we use when we have an imbalance in our data. We will be explaining the metrics in the simplest possible way and will also explain when you should go ahead using a particular metric. It will cover some of the common methods to use while dealing with the imbalance and also some complex methods as well. We will also sight the importance of pipelines and tell you about 'Data Leakage', how can you avoid that as well.

Chapter 5 – Words of Wisdom

It will sight some of the important suggestions that we would like to give to our readers.

Most of the people who want to get into this field start with focusing on ‘Python for Data Science’, they would invest around 3 months to learn Python, they would invest time to learn matplotlib, seaborn but this doesn’t help. In the end, they fail to recollect what they have learned because they have just collected information and didn’t apply that. We tend to remember things which we do more often so if you start working on projects, you would get familiar with the most common constructs, most common visualizations that are used and that’s what is required. It might happen that at the start you won’t get a single line of code but you should not lose hope, the Internet is there to our rescue. Search for those code snippets and you would get the answer and this is how people learn, at any point in time when they get stuck, they open the browser and search for that particular thing. This would also help you develop your skills to do a web search as this is also an important skill.

So this chapter would tell you how your approach should be when you are new to this field.

Chapter 6 – Bot Detection

This chapter would deal with an exciting project which is ‘Bot Detection’. The dataset is huge and it is one of the best projects to understand the importance of feature engineering. You would have to do a lot of research around the problem statement so that you could come up with your assumptions. And you would have to go through a lot of pain to understand this project, it isn’t that easy. So you could get in touch with us on Linkedin in case you have any doubts or queries related to any of the stuff mentioned in this book. Also, it won’t be possible to add the entire project in this book, so we would just be covering important points related to the case studies in this book. The entire project could be found on Github, the Github link would be provided wherever required in the footer section.

The Game of Statistics

In this chapter, we would be dealing with what are Assumptions, why are they important and how could you use them when you deal with some real life problems. It would also cover different statistical tests along with their usage.

If statistical tests bother you, trust us by the end of the first chapter you would feel pretty comfortable with statistical tests and you would get a clear idea about where to use which test.

Assumptions -

An important thing to do before dealing with any problem statement is making some assumptions. *Assumption is a thing that is accepted as true or as certain to happen, without proof.* It is a key skill that is ignored by many. People are fascinated by models and hence ignore the importance of making assumptions.

Assumptions could be of two types - technical assumptions or business assumptions.

Technical assumptions:

Many analytical models or algorithms rely on a certain set of assumptions. This means that before using that model or algorithm we should ensure that our data abides by the assumptions made by the model/algorithm. Technical assumptions could be that a model assumes there should be no collinearity in the data being fed, some models might be affected by the missing data and might also be sensitive to outliers.

EDA helps us to explore our data and thereby helps us to know our data well. During EDA, various technical assumptions will be assessed, which would indeed help us to select the best model to use. If we fail to do such an assessment, we might use a model whose assumptions are violated by the data in hand thereby leading to poor predictions.

For example, Linear regression (Ordinary least squares) has a set of assumptions which the data should abide by before we apply Linear Regression on our data. This might also land up to some incorrect conclusion which could have negative implications on the business.

Business assumptions:

Business assumptions play a vital role when it comes to solving real-world problems. These assumptions are not visible but could be understood by having a deep sense of how the business works and what is the problem that we are trying to solve.

It is very much important to have the correct domain knowledge before dealing with data. Having a sound knowledge about the business makes things easier.

Myth

It is believed that the Data Scientist's job is just to build models. And so, Data Science Enthusiast run behind building models as soon as they get the data. But it should be understood that not all problem statements require a model to provide the solution. Many use cases don't require any model building, they just require getting insights from the data and provide a remedy to the problem statement.

Consider that your client is a YouTuber and he comes up with exciting food recipes. During the current Covid-19 situation as most of us are staying indoors, he thinks of coming up with videos on some exotic dishes. He assumes that as most people are indoors and due to lockdown, people avoid ordering food online. So if I come up with videos on some cool desserts, exotic dishes then I would target more audiences and might get more viewers and subscribers.

So, he puts in all his efforts and starts making videos. But to his surprise, things are not going as he expected. He was expecting that after making videos on exotics stuff he would get more viewers, subscribers but in reality, the number of viewers decreased substantially.

In such a case, model building is not required. We need to analyze the data, gather insights from the data, and then come up with a possible solution. We could ask the client to gather information about his subscribers and viewers. We can then explore this information and try to get an intuition about the data.

In our case, the client assumes that his viewers might be interested in exotic dishes but we all know that preparing exotic dishes consume a lot of time, it requires too much effort and also some dishes might require ingredients that are not readily available amid lockdown. It might so happen that his viewers are of a lazy type and they are not at all interested in investing so much time in cooking, they are okay with simple dishes that can be cooked quickly.

These things can be validated once we have data about the preferences of the viewers. So, we should have a relevant set of questions which we could suggest to our client to ask the viewers. Only by asking relevant questions, we would get answers which could help us to unwrap some of the mystery.

From the above use case, we come to know that it isn't only about model building, it's also about understanding the business domain and then suggesting the client ways by which they could gather data, that will help us to provide a solution to the problem statement. Now consider that after your suggestion, the client thinks of releasing a form with certain questions and hopes that the answer to those questions would help us to analyze the data and eventually help in making some decisions.

It is evident, that out of all the viewers the client has, only a few would fill the form and genuinely answer the questions. But many haven't filled the form, so the data which we will get will not be the population data, rather it would be a sample of data.

Terminology alert

Population - A population dataset contains all members of a specified group (the entire list of possible data values). In our case, if we could get the data for all our viewers then the data would be called population data.
Sample - A sample dataset contains a part, or a subset, of a population. The size of a sample is always less than the size of the population from which it is taken. In our case, only a set of viewers would fill the form, so it is a sample data.

So, it is clear that the data we would be getting will be sample data and not the population data. With the sample data in hand, we cannot make a decision about the population data as it is not the entire data. The data collected is a subset of the population.

Why cannot we make decisions based on the sample data, why do we care so much about population?

There are 2 reasons for this -

1. The sample is a subset of the population, so it won't tell us the entire story about the data, and without knowing the entire story we cannot make decisions.
2. The sample might be biased, there is a possibility that out of all the viewers who have filled the form, most of them feel that exotic dishes are time-consuming but there could be many viewers who are interested in exotic dishes but they didn't want to submit the form and answer the questions.

So, the sample might be biased, and there is a lot of information that the sample might have possibly missed, so just by using the sample, we shouldn't make decisions.

Now it is clear that just by analyzing the sample we cannot come up with conclusions for the entire population.

Our goal is to make decisions based on sample data and extrapolate them to the population data or the data which is not in hand but might come in the future.

We are trying to draw inference about the population by using the sample data. This is where Hypothesis testing comes into the picture.

Hypothesis testing

Hypothesis testing is one of the two methods of Inferential statistics (Confidence Interval is another). In inferential statistics, we take a sample of data from the population and then calculate a statistic (it can be anything like mean, standard deviation, etc.). We then use the value of the statistic to infer (estimate) the value of the corresponding parameter.

A **parameter** is to the population as the **statistic** is to a sample. When we calculate mean, standard deviation, etc., of a population it's called a **parameter**, when calculated for a sample it's called **statistic**.

Terminology alert

Hypothesis:

A hypothesis is a proposition made as a basis for reasoning, without any assumption of its truth.

There's a reason for underlining certain words and you would know its reason as we proceed. There are two types of hypothesis - Null hypothesis and Alternate hypothesis.

Null Hypothesis:

Let us try to understand this with the help of an example. We want to check whether or not there is a difference between the average income of Indian employees in the years 2019 and 2020. So as the word **null** means **zero** or **no**, the null hypothesis would be there is **no difference or zero difference** between the average income of the 2 years.

The null hypothesis can also be a proposition that has been made earlier and that proposition is accepted.

Mathematically -

The average income for the year 2019 = The average income for the year 2020

Alternate Hypothesis:

The alternate hypothesis would say that there is a difference between the two values (which value is larger and which value is smaller is a different question but there is a difference).

In our above example, it would be that both the incomes differ.

Mathematically:

The average income for the year 2019 ≠ The average income for the year 2020.

An alternate hypothesis can also be a statement that differs from the proposition that is accepted by the people.

Real-life example

It has been proposed by Trump that anti-malaria tablets could cure Covid-19. This proposition would become our null hypothesis. It has not been proved that it cures Covid-19, it is only proposed and there is no significant proof of it.

Now a researcher comes up and says that no, the anti-malaria tablet doesn't cure Covid-19. This would become our alternate hypothesis.

When we deal with the hypothesis, we never prove that a hypothesis is correct we just prove that another hypothesis is incorrect.

Similar to our judicial system, a person is innocent until proven guilty. If we cannot prove a hypothesis wrong, it is accepted until it is proved incorrect.

The researcher would have to come up with significant results to prove his proposition. If he fails to do so we would accept that anti-malaria tablets cure Covid-19.

Application in Machine Learning

Let us see how this concept is used in our field. Suppose that we have to build a Linear regression model and we know that the linear regression model would have some features and weights (parameters) assigned to those features. How do we come to know that this model is helpful? This is where **Hypothesis-Testing** comes to our rescue.

Linear regression's equation is -

$$Y = \beta 0 + \beta 1x1 + \beta 2x2 + \ldots.. + \beta nxn$$

The null hypothesis says that the model has no effect, meaning $\beta 0 = \beta 1 = \beta 2 = \ldots\ldots \beta n = 0$. And this is what the null hypothesis will always claim, it will claim that our model is useless.

An alternate hypothesis would be that the model with these parameters has some effect and is better than the model with beta values equal to zero.

Mathematically, $\beta 0, \beta 1, \beta 2 \ldots\ldots \beta n \neq 0$.

We will have to prove that this model is significant using some hypothesis tests. If we fail to reject the null hypothesis, then our model is useless.

Terminologies

Let us go through some of the important terminologies related to Hypothesis testing.

Reject Null hypothesis:

This means that we have proved that the null hypothesis is wrong and we go ahead with an alternate hypothesis. But we never say we accept the alternate hypothesis. We always talk in terms of the Null hypothesis and hence we say that we Reject Null Hypothesis.

This term implies double negation, *reject* is one level of negation and *null* is the second level of negation.

Fail to reject the Null hypothesis:

This means that the null hypothesis is correct and alternate is wrong, it is clear from above that we won't say we reject the alternate hypothesis. The definition of a hypothesis is, Hypothesis is a proposition made as a basis for reasoning, without any assumption of its truth. **We don't prove the truthfulness of a hypothesis, we just prove that it's incorrect** hence, we have underlined certain words. So, we never accept the null hypothesis as we don't prove its truthfulness, we say that we Fail to Reject it as we don't have any strong evidence against it.

This term implies triple negation, *fail* is one level of negation, *reject* is the second level, and *null* is the third level of negation.

P-value:

Since long we have been talking about significance, to reject the null hypothesis we need to prove the significance, and to prove significance we perform some statistical test which gives us a p-value. P-value is nothing but the probability value.

*The **p-value** is the probability of obtaining results as extreme as the observed results of a statistical hypothesis test, assuming that the null hypothesis is correct.*

This p-value helps us to prove significance. We will also look at how to calculate this p-value in the upcoming part of this book. We set a certain threshold and if the p-value is less than the threshold we reject the null hypothesis.

Generally, the threshold is set to 0.05, meaning that there is a 5% chance that we make a wrong decision by rejecting the null hypothesis, when we shouldn't be rejecting it. It means that if we reject the null hypothesis 100 times for a p-value less than 0.05 then only 5 times our decision might be wrong, rest 95 times it would be correct.

Why do we reject the null hypothesis when the p-value is less than 0.05?

The hypothesis test that we perform assumes that the null hypothesis is true. If the null hypothesis is true then we should get a high probability. If we are getting very less probability, less than 5% then we are pretty sure that our assumption that the null hypothesis is true is incorrect, hence we reject null hypothesis when the p-value is less than 0.05. *P-value is also the probability that what we have seen (in the above example that coefficients are non-zero) is due to random chance.* If the probability is small then we are sure that the change is not due to random chance, so we reject the null hypothesis.

Note: The threshold is generally 0.05 i.e. 5% and we can reduce it to 1% as well if we want to reduce the chances of making an incorrect decision. For example, if we are performing a hypothesis test for a drug then we need to reduce the chances of making an incorrect decision. In such cases, we might keep the threshold as 0.01 or 1%.

Hypothesis testing at its core checks whether our statistic belongs to the null hypothesis distribution or some other distribution. If it does not belong to our null hypothesis distribution, we say that our statistic comes from some other distribution and we reject the null hypothesis.

This is how statisticians talk, if you are able to understand the below image then you can say that you have understood the terminologies well. In case if you don't, read the chapter again and we hope that you would understand it.

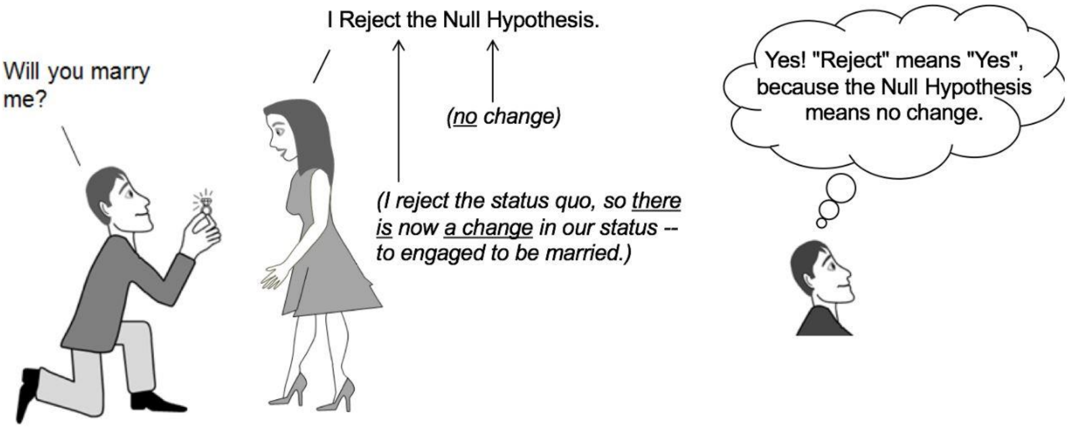

Will you marry me?
I Reject the Null Hypothesis.
(no change)
(I reject the status quo, so there is now a change in our status -- to engaged to be married.)
Yes! "Reject" means "Yes", because the Null Hypothesis means no change.

Figure 3: Conversation between statistiticians[1]

Now, let us have a look at some of the statistical tests and also get an understanding of their usage.

But before we dive into the statistical tests, it is important to understand how data is represented and different levels of measurements as it plays a key role in identifying which statistical test to use.

[1] Image Source - https://miro.medium.com/max/2920/1*0T7xtPuohhs7VJl9CfLYvQ.png

Levels of Measurement

The way a set of data is measured is called it's level of measurement. There exist four levels of measurement. Nominal, Ordinal, Interval, or Ratio (Interval and Ratio levels of measurement are sometimes called Continuous or Scale).

Why is it important?

We need to understand the different levels of measurement, as how the research question is phrased together with the levels of measurement, dictates which statistical test is appropriate.

Types of Levels of Measurement

Nominal - Data categorized using this scale is qualitative, as the name implies, they refer to names or labels without specifying any order. They don't have a numeric value and neither can be added, subtracted, divided, or multiplied. If they appear to have an order then you probably have ordinal variables instead.

Example - Car brands like Tata, Hyundai, Honda, Suzuki, etc.

There is no specific order and no brand is superior to the other.

Ordinal - Similar to nominal but with an inherent order. The difference between the 2 ordinal variables need not be the same as the difference between 2 other ordinal variables.

Example - Let us consider the top 10 countries worst hit by Corona, now here, we have some ordering between the countries. The difference in rank 2nd and 3rd is 1 and same is the difference in rank 5th and 6th. But the difference in the number of cases between 2nd and 3rd country might be different than the difference in the number of cases between 5th and 6th worst-hit country. So we say that the difference between the 2 ordinal variables need not be the same as the difference between 2 other ordinal variables.

Interval - An interval scale has ordered numbers with meaningful divisions. The interval scale of measurement not only classifies and orders the measurements, but it also specifies that the distances between two intervals are equal.

Example - Temperature measured in centigrade would fall under the interval scale as the difference between 20 and 30 degrees centigrade would be the same as the difference between 50 and 60 degrees centigrade. This is because each 10-degree interval has the same physical meaning (in terms of the kinetic energy of molecules)

Ratio - It is an interval scale with the additional property that it's zero position indicates the absence of the quantity that is being measured. We can think of a ratio scale as the combination of the other three scales.

Like a nominal scale, it will provide a name or category for each object (the numbers serve as labels).

Like an ordinal scale, the objects are ordered (in terms of the ordering of the numbers).

Like an interval scale, the same difference at two places on the scale has the same meaning. And also, the same ratio at two places on the scale carries the same meaning.

Example - Weight of 0 (zero) units means that there is an absence of weight but temperature of 0 (zero) degrees centigrade doesn't imply the absence of temperature (or 0 kinetic energy of molecules)

Sampling Techniques

Before starting with sampling let's revisit what is a population and sample.

Population: According to Wikipedia, a population is a set of similar items or events which is of interest for some question or experiment.

To understand it better let's take an example, suppose we want to find out what is the average salary of Data Scientists in India. So, all Data scientists in India become our population or we can call it a subject of study.

Sample: A sample is a set of individuals or objects collected or selected from a statistical population.

Our population at study is a set of all Data Scientists in India, but we have limited time and resources. We cannot go and knock the door of every Data Scientist in India and ask their salary. So, we randomly choose some observations from our population.

Let’s say we randomly choose 100 Data Scientists and get their salary details, so this becomes our sample with 100 observations, while the population is all the Data Scientists in India.

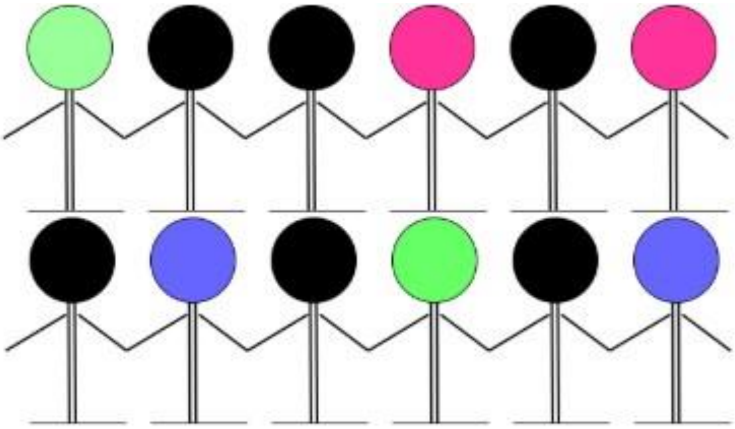


Figure 4: Population - All Data Scientists

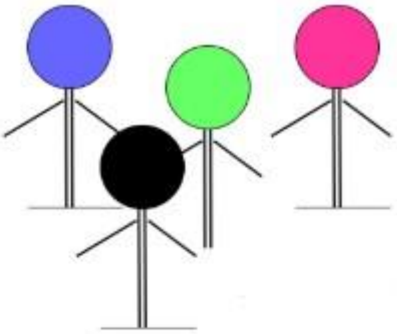


Figure 5: Sample - Subset of All Data Scientists (Population)

A sample should be chosen such that it should be representative of a population.

This means that the sample should have almost all the characteristics of a population. The distribution of the sample should be the same as that of the population. For example, in a population, if class A has 1000 observations and class B has 500 observations this means the ratio of observations of class A to class B is 2:1 so the same ratio should be maintained in the sample as well.

Different Sampling Techniques:

There are two types of sampling techniques -

1. **Biased sampling** - leads to a biased sample.
2. **Unbiased sampling** - leads to an unbiased sample.

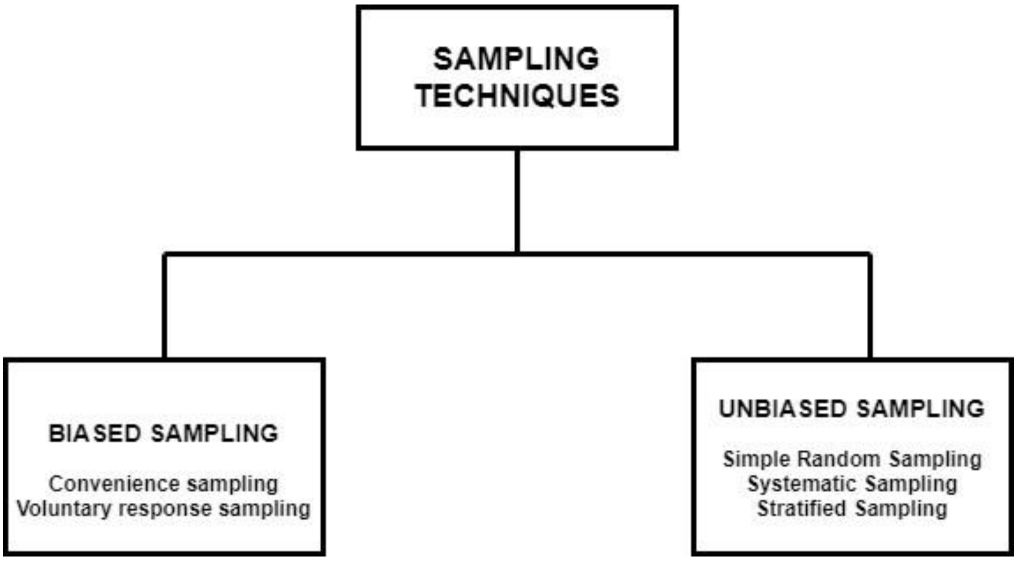


Figure 6: Sampling Techniques

Biased Sampling:

Biased Sampling is the worst method of all the Sampling methods since it doesn’t give an equal chance to all the observations in the population. It occurs when some set of observations of the population are favored over others.

They are of 2 types:

Convenience Sample - Only Includes people who are easy to reach.

For example, if we want to receive feedback from 5 customers and we have around 100 customers. So, let’s say that we select our first 5 customers or customers whose name starts with a particular alphabet. Here not every customer is getting a chance so we call it *Convenience Sample*. The observations of the sample were chosen as per our convenience.

Voluntary Response Sample- Consists of people who have nominated themselves out of their own will.

For example, if we ask which all customers are interested in filling the feedback form and around 5 customers are ready to do so on their will, then it becomes a *Voluntary Response Sample*. Those with a strong interest are the ones who are most likely to fill the feedback form.

A good sample is the one that is representative of the entire population and it gives an equal chance of being chosen.

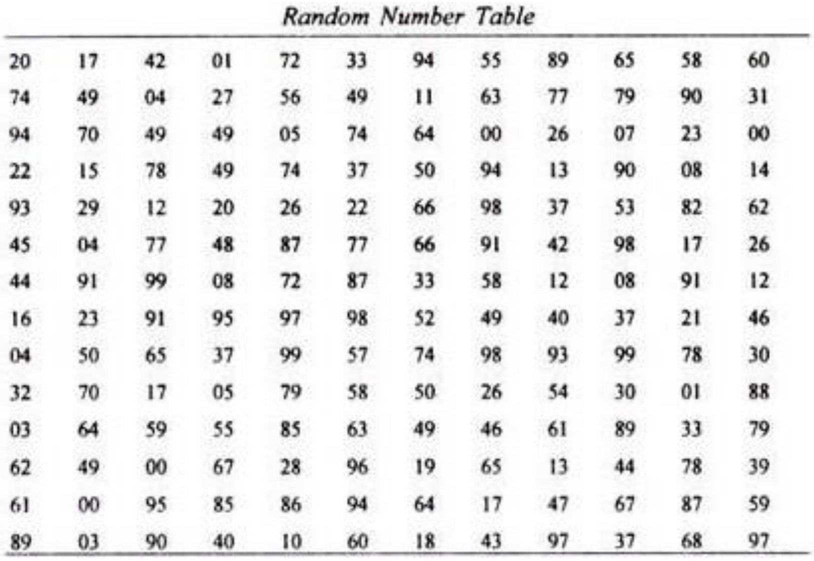
Unbiased Sampling:

Unbiased Sampling is the best of all the Sampling methods since it gives equal chance to every observation in the population.

Different types of Unbiased Sampling methods:

Simple Random Sampling (SRS) - SRS is a sampling technique where every item in the population has an equal chance or likelihood of being selected.

For example, we could randomly choose customers and ask them to fill the form. We could also use a random number table and then choose our observations accordingly. So, in this way, all customers will get an equal chance of being chosen with no bias*.



Random Number Table

20	17	42	01	72	33	94	55	89	65	58	60
74	49	04	27	56	49	11	63	77	79	90	31
94	70	49	49	05	74	64	00	26	07	23	00
22	15	78	49	74	37	50	94	13	90	08	14
93	29	12	20	26	22	66	98	37	53	82	62
45	04	77	48	87	77	66	91	42	98	17	26
44	91	99	08	72	87	33	58	12	08	91	12
16	23	91	95	97	98	52	49	40	37	21	46
04	50	65	37	99	57	74	98	93	99	78	30
32	70	17	05	79	58	50	26	54	30	01	88
03	64	59	55	85	63	49	46	61	89	33	79
62	49	00	67	28	96	19	65	13	44	78	39
61	00	95	85	86	94	64	17	47	67	87	59
89	03	90	40	10	60	18	43	97	37	68	97

Figure 7: Random Number table[1]

* there is nothing which is free from bias, some bias in present unintentionally, we aim to have minimal bias.

[1] Image Source - https://research-methodology.net/wp-content/uploads/2015/04/Simple-random-sampling.png

Systematic Sampling: Systematic random sampling is the random sampling method that requires selecting samples based on a system of intervals in a numbered population.

For example, from all the customers we decide to call every 3rd customer for feedback. So, we arrange them systematically and choose every 3rd customer for the feedback.

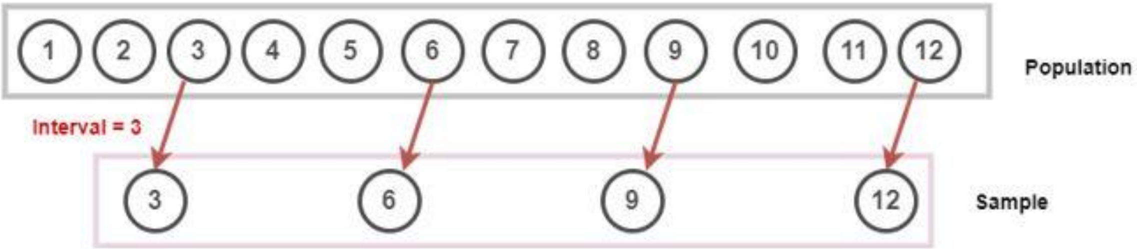


Figure 8: Choosing every 3rd observation

Stratified Random Sampling - With stratified sampling, the population is divided into separate groups, called "strata". Then, a probability sample (often a simple random sample) is drawn from each group.

Let's say we want to study the number of males and females in the Data Science field, and somehow, we know that the number of males and females is not present in equal ratio. Let's say we know the ratio is 80:20 so this is the information we have about your population i.e., our subject of study. Now when we go for sampling, we will try to replicate this ratio in our sample as well i.e., we will try to keep the ratio of 80:20 in our sample as well.

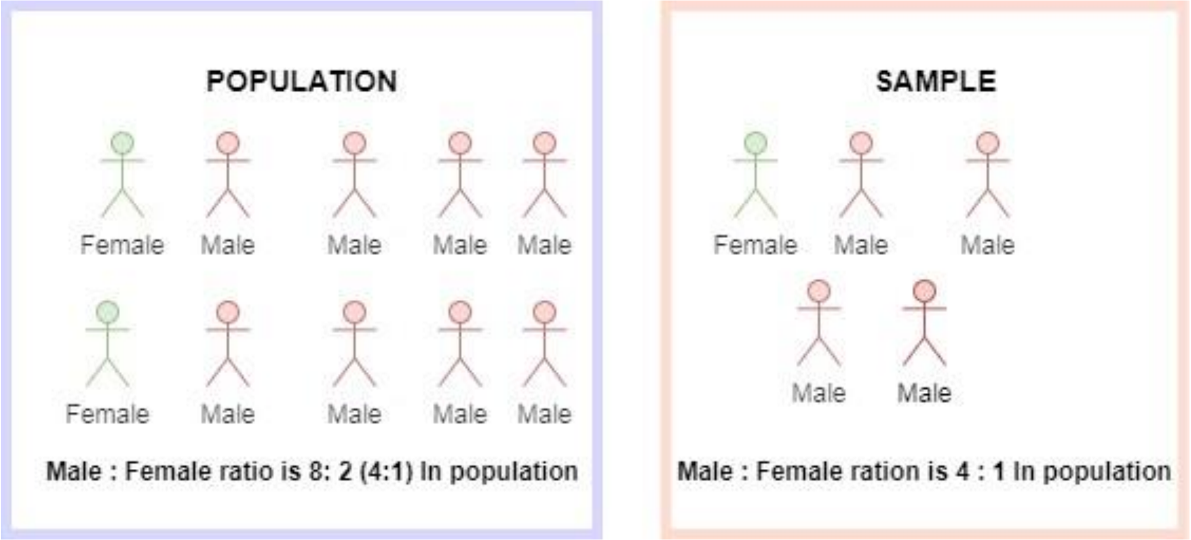

POPULATION
Female
Male
Male
Male
Male
Female
Male
Male
Male
Male
Male : Female ratio is 8: 2 (4:1) In population
SAMPLE
Female
Male
Male
Male
Male
Male : Female ration is 4 : 1 In population

Figure 9: The ratio is maintained in the sample

Using Stratified Sampling we could reduce the sample size as it would follow a distribution similar to the population. Hence, even after using lesser data we have similar information about data.

So, with this knowledge in hand, we can get started with Statistical tests.

Statistical Tests

All the statistical tests are divided into two types. They could either be **one-tailed** or be **two-tailed**. So, the category it falls into depends on the problem statement and the hypothesis we set. Let us try to understand this with the help of a simple example.

One-Tailed tests

Suppose you run a potato chips company and you claim that the packet of chips weighs not less than 250 gms but the customers claim that the weight is less than 250 gms.

The null hypothesis would be - ***Weight >= 250 gms***

And the alternate hypothesis would be - ***Weight < 250 gms***

The one-tailed test could be left-tailed or right-tailed depending on the formulated hypothesis

Two-Tailed tests

Consider the same example that we mentioned above but now let's say that you are claiming that the weight of chips is exactly 250 gm and the customer's claim that there is a difference, it might be less than 250 gms or it might be more than 250 gms but it is not exactly equal to 250 gms.

The Null hypothesis would be - ***Weight = 250***

And the alternate hypothesis would be - ***Weight ≠ 250***

The image on the next page depicts a One-tailed test and a Two-tailed test. The two images at the top of the image represent a Left tailed test and Right tailed test. The image below depicts a Two-tailed test.

The area shaded in blue color is the critical region, if our p-value falls in the critical region then we reject the Null hypothesis.

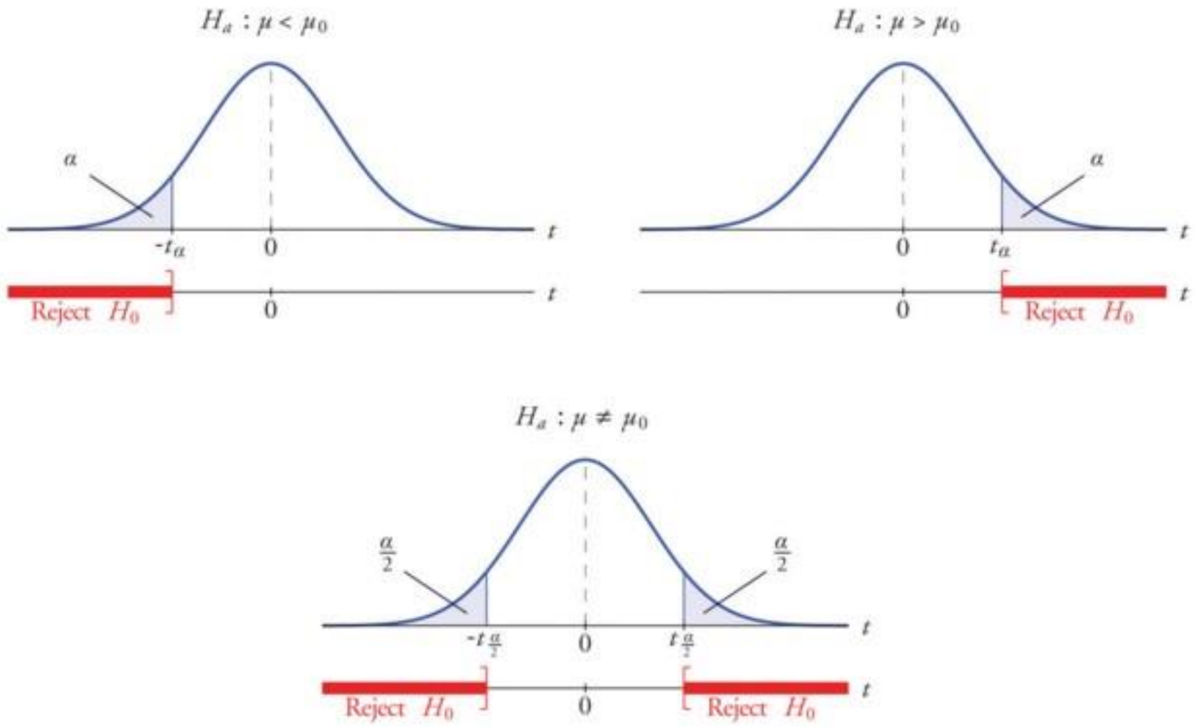


Figure 10: One tailed and two tailed test[1]

To be in a position to identify which statistical test to use we should know two things in advance:

- Does the data meet a set of assumptions (because the statistical tests come with a set of assumptions about the data)?
- Nature of the data and types of the variables we are dealing with.

Knowing these two things makes it easier for us to decide which test to use.

Now what all assumptions could be made by a statistical test, let us have a look at it.

Assumptions about data

Assumptions about Variance -

Some tests assume that different groups within the data have similar variance or a statistical way of saying the same thing is, **the groups should not have a significant difference in the variance.**

[1] Image Source - https://miro.medium.com/max/619/1*Zu0iou9DD-zIZSOZjsUeEA.png

So, we should check that the test we think of applying doesn't assume the variance to be similar, in case if it assumes then we need to run a statistical test to check the variance, and then we should apply our intended statistical test.

Assumptions about normality -

Most of the statistical tests assume that the data follows a normal distribution.

Again, the statistical tests could be of two types, one which has some set of assumptions about the data and the ones which don't. Tests that have a set of assumptions about the data (in other words the data should adhere to some set of assumptions) are called **parametric tests.** The ones which don't have any assumptions are called **non-parametric tests.**

The choice of the statistical test also depends on the type of data in hand.

Types of data

Depending on the data type, it could be divided into two types-

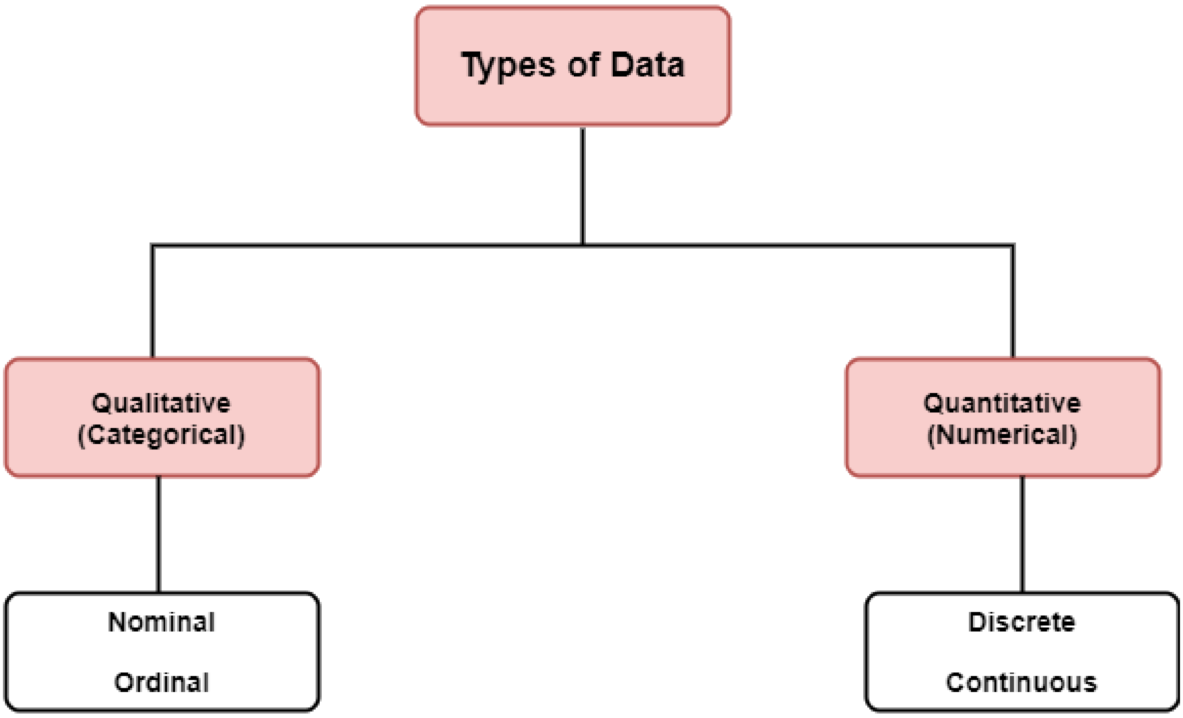


Figure 11: Types of Data

Quantitative data - Represents values or counts of an observation, generally numeric.

- **Discrete data** - It can only take certain values and might not take the infinite time or forever to count.
  Example - Number of bikes in a city. We cannot have half a bike.

  We cannot split discrete data further, in some cases, we can but, that would convert it into continuous data.

- **Continuous data** - It can have any value in a range and might take the infinite time or forever to count.
  Example - time could take any value like 2 hours, 2 mins, 3 secs, 6 microseconds, 8 nanoseconds, etc. So, time is a continuous variable.

  It can be made discrete and then we could measure it like time in hours, time in seconds.

Categorical data - Represents a group of observations

- **Binary** - Has only 2 categories (yes/no, male/female)
- **Ordinal** - represents an order (1st, 2nd 3rd)
- **Nominal** - More than 2 categories (names of cities)

Let us first have a look at Parametric tests -

As mentioned above, these tests have a set of assumptions which the data should adhere to.

Parametric tests -

It could be used to compare means and variances.

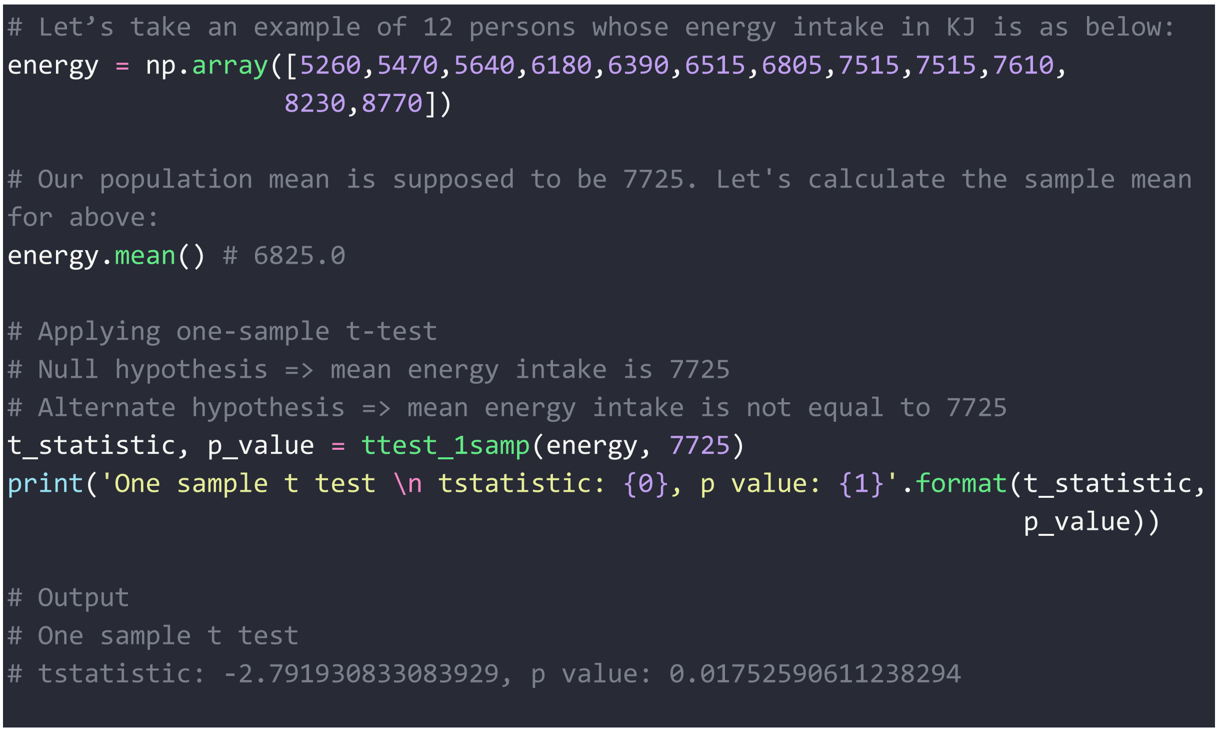
Comparing means -

The T-test is used to compare means, there are many variants of the T-test. ANOVA is also used to compare means and it also has different variants.

One sample Independent T-test -

This test is used to check if there is a significant difference between the sample mean and the population mean. It assumes that data comes from a normal distribution, it is used when we have 1 sample under observation. But if the sample size is large enough then we don't care about the assumption of normality as well (more about this in Central Limit Theorem section).

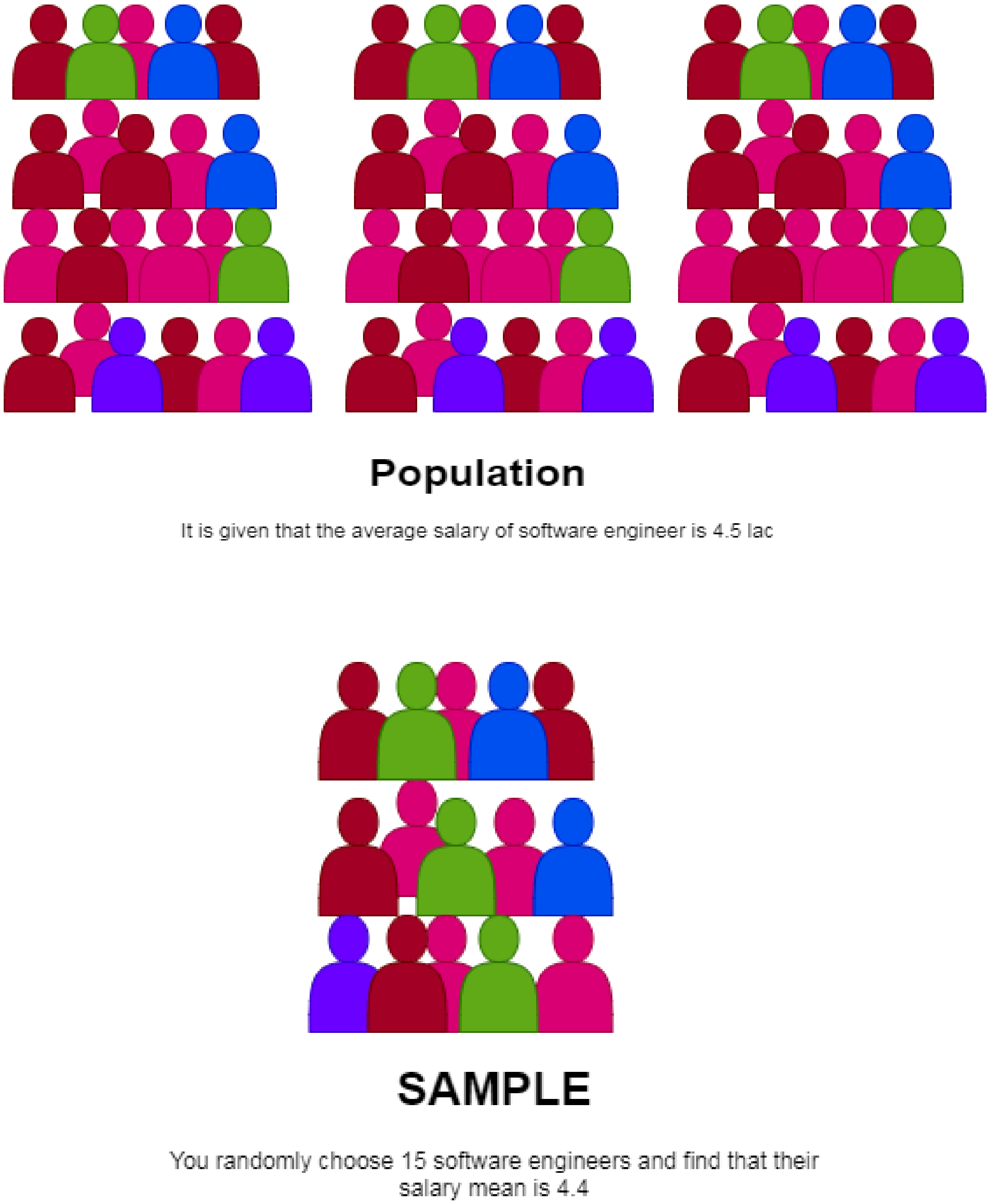
Example - Suppose if we want to check if the energy intake of 12 people is significantly different from a theoretical value, we could use t-test (assuming the required assumptions are adhered by the data)



```
# Let’s take an example of 12 persons whose energy intake in KJ is as below:
energy = np.array([5260,5470,5640,6180,6390,6515,6805,7515,7515,7610,
                   8230,8770])

# Our population mean is supposed to be 7725. Let's calculate the sample mean
for above:
energy.mean() # 6825.0

# Applying one-sample t-test
# Null hypothesis => mean energy intake is 7725
# Alternate hypothesis => mean energy intake is not equal to 7725
t_statistic, p_value = ttest_1samp(energy, 7725)
print('One sample t test \n tstatistic: {0}, p value: {1}'.format(t_statistic,
                                                                  p_value))

# Output
# One sample t test
# tstatistic: -2.791930833083929, p value: 0.017525906112382
```

If you want to check if the average pay for Software engineers is 4.5 lac and you have a group of 15 friends with the same designation, their average pay comes out to be 4.4 lac.

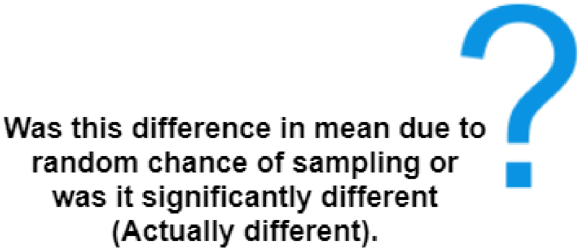
So, the group of 15 friends becomes your sample, the sample mean is 4.4 and the population mean is 4.5. So, you could run an Independent T-test to check if there is a significant difference between the two means.



It is given that the average salary of software engineer is 4.5 lac

You randomly choose 15 software engineers and find that their salary mean is 4.4

Figure 12: Population and Sample



Was this difference in mean due to random chance of sampling or was it significantly different (Actually different).

Figure 13: Think about it

Two samples Independent T-test -

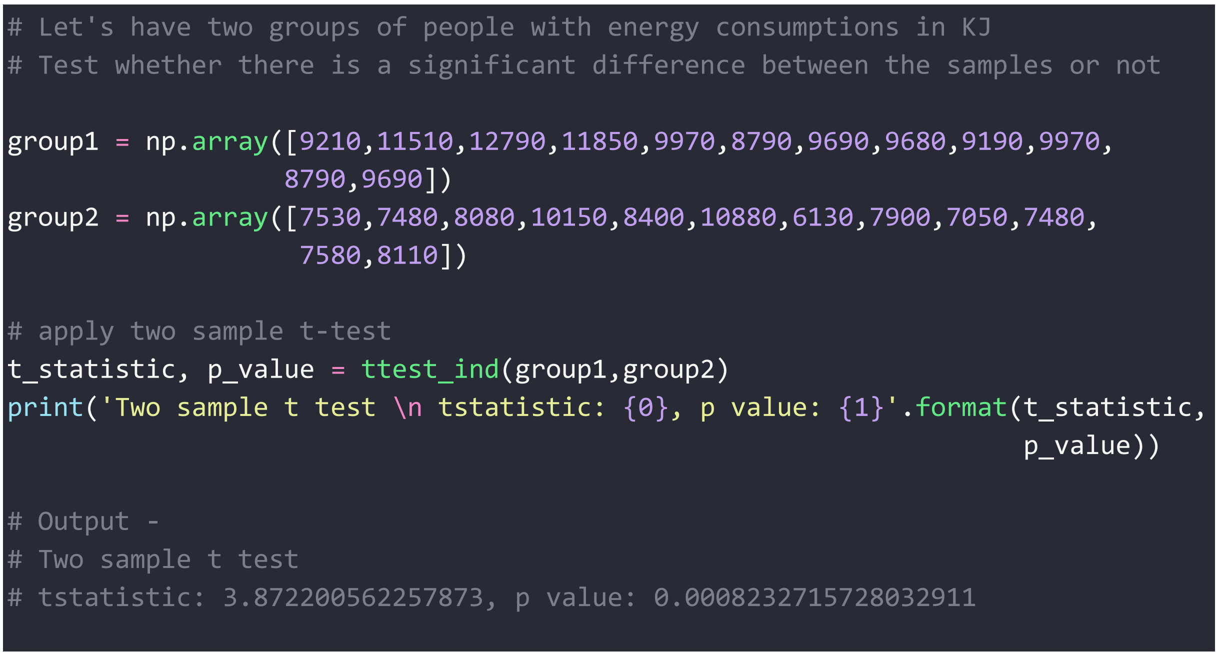
This test is used to check if there is a difference between the mean of 2 samples. Similar to One sample t-test, this test also has similar assumptions.

Example - If we want to compare the salaries of employees from 2 different companies, we would choose 2 samples (one from each company) of around 15 observations and then would check if there is a significant difference between the mean of the 2 samples.

Also, we could check if the mean of two groups of people has a significant difference in their energy consumption or not?

H0: mean energy intake of the two samples is equal

Ha: mean energy intake of the two samples is not equal



```
# Let's have two groups of people with energy consumptions in KJ
# Test whether there is a significant difference between the samples or not

group1 = np.array([9210,11510,12790,11850,9970,8790,9690,9680,9190,9970,
                  8790,9690])
group2 = np.array([7530,7480,8080,10150,8400,10880,6130,7900,7050,7480,
                   7580,8110])

# apply two sample t-test
t_statistic, p_value = ttest_ind(group1,group2)
print('Two sample t test \n tstatistic: {0}, p value: {1}'.format(t_statistic,
                                                                  p_value))

# Output -
# Two sample t test
# tstatistic: 3.872200562257873, p value: 0.0008232715728032911
```

Since the p-value is less than 0.05, we can reject the null hypothesis at 5% level of significance.

This means that there is a significant difference between the two means.

Paired T-test -

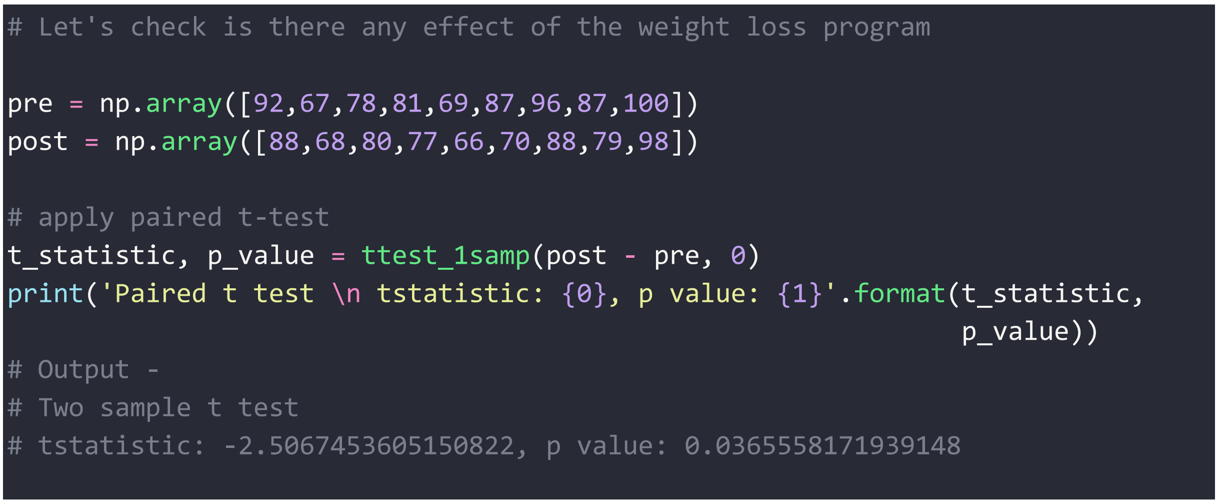
The use of this test could be best understood with the help of an example.

Consider that you own a fitness company and you want to check if the weight loss program that you have launched is actually beneficial.

To check this, you pick 20 people at random and calculate their average weight and it comes out to be ‘X’, post the weight loss program you calculate the average weight of the same people and call this as ‘Y’.

Now you check if there is a significant difference between X and Y, if yes then the weight loss program of yours has actually helped.

This means that we use Paired T-test when we want to compare means of 2 samples from the same population (People under observation are the same, hence population is the same but their weight values have changed hence we say that we are dealing with 2 different samples).



```
# Let's check is there any effect of the weight loss program

pre = np.array([92,67,78,81,69,87,96,87,100])
post = np.array([88,68,80,77,66,70,88,79,98])

# apply paired t-test
t_statistic, p_value = ttest_1samp(post - pre, 0)
print('Paired t test \n tstatistic: {0}, p value: {1}'.format(t_statistic,
                                                               p_value))
# Output -
# Two sample t test
# tstatistic: -2.5067453605150822, p value: 0.0365558171939148
```

Since the p-value is less than 0.05, we can reject the null hypothesis at 5% level of significance.

This means that there is a significant difference between the weights before and after the weight loss program.

ANOVA -

T-tests are used when we want to compare 2 means, but what if we want to compare more than 2 means, in such a case ANOVA comes to our rescue. We use ANOVA when we want to compare more than two means.

ANOVA (Analysis Of Variance), the name is quite misleading and so people get carried away by its name. Some people assume that ANOVA is used to analyze variance and which is WRONG. We know it is the name which brings us to such a conclusion but the fact is ANOVA assumes that there is no significant difference between the variance of the groups/samples. So here is the catch, before proceeding with ANOVA we need to check if the variance is similar and to do this, we have tests which compare the variances. Also similar to the t-test, ANOVA also assumes the normality of data.

Comparing Variances

Fisher's test -

This test could be used to compare the variance of 2 samples coming from a population that is normally distributed. It could be used to validate the assumption related to similar variance before applying ANOVA or any other test.

Correlation tests

Pearson's Correlation Coefficient -

This test is used to check the association between quantitative variables. This test also has a set of assumptions like no outliers should be present in the variables under observation, the presence of a pair of values. For example, if this test is applied on weight and height variables then each observation used should have value for weight and height variable. Some more assumptions should be adhered to before using this statistical test.

Non-Parametric tests -

There are scenarios where it becomes quite difficult to adhere to the assumptions made by the parametric tests. This is where non-parametric tests come to our rescue but one thing worth noting is that we could come up with more strong conclusions using parametric tests than non-parametric tests.

Reasons for using Non-Parametric tests

- **Non-normal or skewed distribution -**

  Many times, our data might not come from a normal distribution and the distribution might be skewed (obviously we would try to apply some transformation to convert our skewed data into normal distribution but this doesn't help at all times).

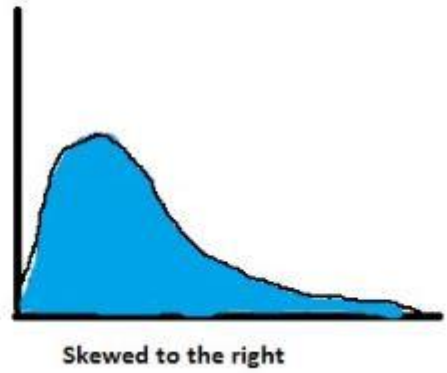
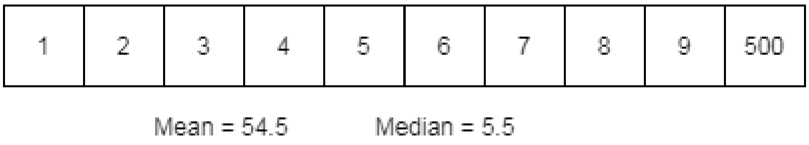


Figure14: Data skewed to the right

- **Median is better than mean -**

  As the data in hand would be skewed, the mean would not be a good approximator of central tendency. We would go with the median and we could be interested in comparing the medians rather than the means as medians are not much affected by outliers than the means.

  For instance, consider we have 10 observations:



1	2	3	4	5	6	7	8	9	500

Mean = 54.5 Median = 5.5

Figure 15: Image showing impact of outliers on mean and median

The mean of the observations come out to be 54.5 and the median is 5.5, in this case, the mean is not the appropriate measure of central tendency. So, we take the median as a measure of central tendency.

- **Ordinal data or outliers -**

  The data which we would like to investigate might be ordinal and might be prone to outliers that we can't get rid of. In such a case, we prefer non-parametric tests.

Comparison medians -

Wilcoxon signed-rank test -

In most cases, it is used as a possible alternative to Paired T-test when the data isn't following the assumptions mentioned by the t-test.

This test aims at comparing median to a hypothesized median as when the data is skewed or contains outliers, the median is preferred over mean.

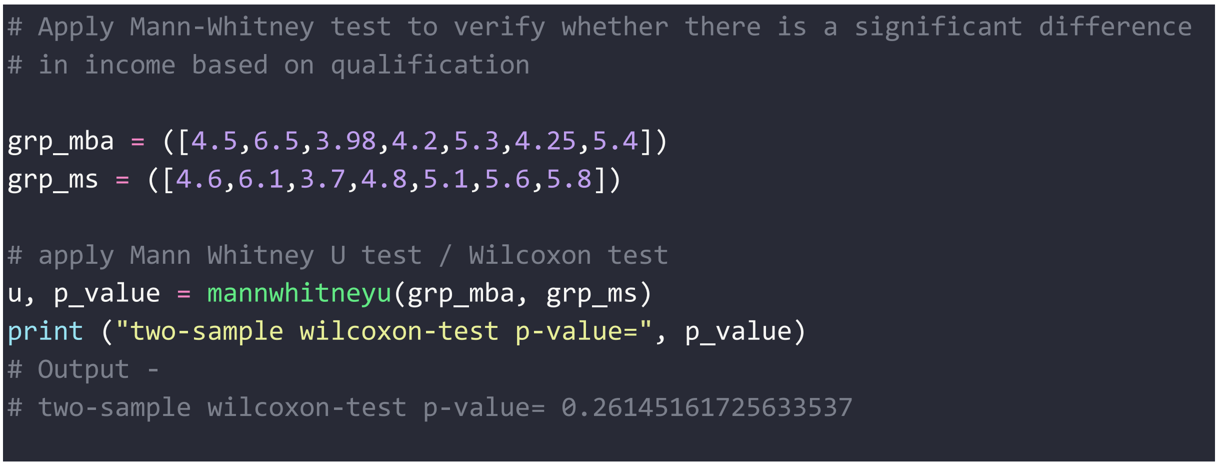
Kruskal Wallis test -

This test is used when the independent variable has more than 2 categories/groups and the dependent variable is ordinal or continuous. It is used to compare medians when the data isn't normally distributed and. It is also sometimes used as a non-parametric version of ANOVA and unlike the Mann-Whitney U test, it allows more than 2 groups/categories to be compared.

Wilcoxon rank-sum test -

This test is also known as the Mann-Whitney U test. It is used to compare the medians of two independent groups when the dependent variable is not normally distributed and is ordinal or continuous. This is used as an alternative to Independent sample T-test though that isn't the case always.

Example - Suppose we want to compare the pay based on educational level, the independent variable in this case would be the educational level which would have 2 categories like MBA, MS. The dependent variable would be continuous that is the pay, Mann-Whitney U test could be used in this scenario. The pay data might be skewed or non-normal.



```
# Apply Mann-Whitney test to verify whether there is a significant difference
# in income based on qualification

grp_mba = ([4.5,6.5,3.98,4.2,5.3,4.25,5.4])
grp_ms = ([4.6,6.1,3.7,4.8,5.1,5.6,5.8])

# apply Mann Whitney U test / Wilcoxon test
u, p_value = mannwhitneyu(grp_mba, grp_ms)
print ("two-sample wilcoxon-test p-value=", p_value)
# Output -
# two-sample wilcoxon-test p-value= 0.26145161725633537
```

Correlation tests –

Spearman's Correlation -

It is used when one or both the variables aren't normally distributed and mostly used as an alternative to Pearson's Correlation test. It is mostly used when the data is ordinal. It calculates the strength and direction of the association between the 2 variables.

Chi-square test -

It is used to check the association between two categorical variables. The frequency of each category for one categorical variable is compared across the categories of the other categorical variable. The data is represented using a contingency table where each row represents categories of one variable and each column represents categories of the other variable. The cells contain the corresponding frequencies of the categories.

For Chi-square test

Null hypothesis - Both the variables are not associated

Alternate hypothesis - Both the variables are associated

Significance level = 0.05

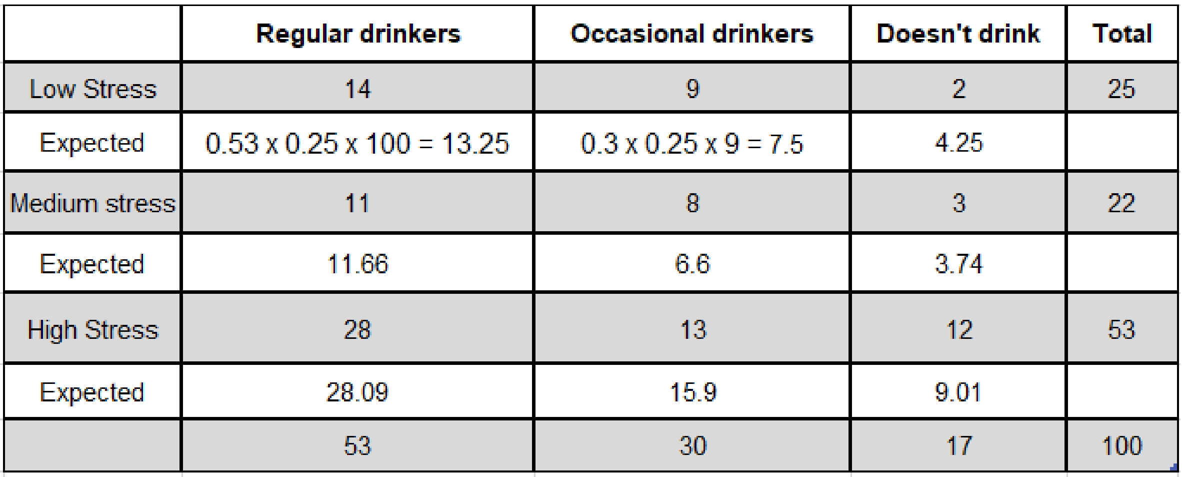
Let us try to understand this with the help of an example -

We will try to identify is there is an association between stress level and drinking habits.

	Regular drinkers	Occasional drinkers	Doesn't drink	Total
Low Stress	14	9	2	25
Medium stress	11	8	3	22
High Stress	28	13	12	53
	53	30	17	100

From the above table, we can see that we have 100 observations, and 25% of people have low stress, 53% of people are Regular drinkers (assuming the null hypothesis is true).

So, the number of observations with Regular drinkers and low stress is 0.53 x 0.25 x 100 = 13.25. This is the expected number of observations if the variables are not associated. Similarly, we will calculate for the rest.



	Regular drinkers	Occasional drinkers	Doesn't drink	Total
Low Stress	14	9	2	25
Expected	0.53 x 0.25 x 100 = 13.25	0.3 x 0.25 x 9 = 7.5	4.25	
Medium stress	11	8	3	22
Expected	11.66	6.6	3.74	
High Stress	28	13	12	53
Expected	28.09	15.9	9.01	
	53	30	17	100

$$Chi - square\ statistic = \sum \frac{(Observed - \ Expected)^2}{Expected}$$

$$= \frac{(14-13.25)^2}{13.25} + \ldots\ldots + \frac{(12-9.01)^2}{9.01}$$

$$= \ Calculated\ value$$

We will get the calculated value after we do the calculations and this value will be compared to the value we obtain from the chi-square table. But to obtain value from the table, we need degrees of freedom (dof).

Let us try to understand degrees of freedom with the help of a small example, suppose you have to find 3 numbers such that their sum is 10, call these numbers as a, b and c. We are free to choose any value for variable ‘a’, consider we chose it as 3, similarly we could choose any value for variable ‘b’ and suppose we chose a value for ‘b’ as 5 then we are not free to choose any value we wish for variable ‘c’. As soon as we fix the values for ‘a’ and ‘b’ we are bounded by constraints and hence the value of ‘c’ comes out to be 2 so that the sum comes up as 10.

We had 3 variables and we were free to vary the value of 2 variables so in this case, we could say we have 2 degrees of freedom i.e. no. of variables – 1. Similarly, we calculate degrees of freedom below -

$$
\begin{aligned}
\textit{Degrees of freedom (dof)} &= (\textit{no. of cols} - 1) \text{ x } (\textit{no. of rows} - 1) \\
&= (3 - 1) \text{ x } (3 - 1) \\
&= 4
\end{aligned}
$$

So, in the Chi-square table, we would check for a value corresponding to significance as 0.05 and dof as 4. Consider the value we get is called “*table value*”, the *calculated value* would be compared with *table value* and accordingly we would *reject the null hypothesis* or *fail to reject the null hypothesis.*

The chi-square test could be used when the expected frequency is greater than 5, in our case the expected values were greater than 5 hence we could easily use the chi-square test. But in case if the expected frequency is less than or equal to 5 then we should go for Fisher’s exact test.

Fisher’s exact test -

This test is used when we want to conduct a chi-square test but the frequency is less than or equal to five. We should remember that the chi-square test assumes that the expected frequency should be greater than five but Fisher’s exact test does not have any such assumption and it can be used regardless of how small the expected frequency is.

Pros and Cons of Parametric and Non-Parametric tests.

- Parametric tests are more powerful than non-parametric tests so if we have data which adheres to the assumptions made by a parametric test then without giving any second thought we should go with non-parametric tests.
- But it isn’t always the case that the data would be normal, it might contain outliers that are important. So in such cases, we have to go with non-parametric tests.
- One beauty of statistics is that we tend to care less about the normality assumption as the sample size increases. So, if the sample size is large (generally greater than 30) we could go for parametric tests. The reason for this is the Central limit theorem.

Let us try to understand the Central limit theorem in the next section and see how it helps in hypothesis testing.

Central Limit Theorem (CLT)

The Central limit theorem is one the widely used theorem, it is used quite often but its use is abstract and hence people don't even realize that they are using it. This theorem becomes very important when we are dealing with non-normal distribution and so it is important to have some knowledge around this before performing hypothesis testing.

The definition of the Central Limit theorem might seem quite technical so let us try to understand it by thinking in the following manner. We draw a sample of size 'n' from a population, using the sample we can calculate the sample mean and standard deviation of the sample.

Let's suppose that we do not have any idea about the distribution of the population, we are also not aware of the mean and standard deviation of the population. Assume that the sample size is greater than 30 and we want to apply a parametric test on this sample, but the sample comes from a population whose distribution is let's say non-normal.

Should should we go ahead with a non-parametric test? Hang on for a minute before answering this question.

Let us assume that we have drawn some more samples from the population and have calculated their means (called as sample mean). When we try to plot the sample means, the distribution that we would get would be close to a normal distribution and as we increase the sample size, the distribution is more likely to resemble a normal distribution (a bell-shaped curve). This is what the Central limit theorem tries to convey.

It also conveys that the mean of the distribution obtained from the sample means would be equal to the population mean irrespective of the distribution of the population.

Formal definition -

The Central limit theorem says that the mean of the ***sampling distribution of the sample means*** *is equal to the population mean irrespective of the distribution of the population and when the sample size is greater than 30.*

Let us try understanding the meaning of the highlighted terms, ***sampling distribution*** means that the distribution is made up of samples and the later part i.e. ***sample means*** implies that the distribution is made up of “means of the sample”. We know in the Central limit theorem we create several samples of size greater than 30, calculate the mean of the samples, and then plot them.

Mathematically it states that -

Let **µ** be the population mean and **σ** be the population standard deviation. If we draw a sample of size **N** from the population then according to CLT the mean of the sampling distribution of sample means is given as -

$$\bar{X} = \mu$$

The standard deviation of the sampling distribution of sample means (meaning standard deviation of the distribution of sample means) is given as -

$$\sigma_{\bar{X}} = \frac{\sigma}{\sqrt{N}}$$

Takeaways –

1. The mean of the sampling distribution of sample means is equal to the population mean.
2. Also, the sampling distribution of the sample means follows a normal distribution if the sample size is greater than 30.

So, you might be wondering how is this going to help us. Suppose if we run a t-test, it assumes that the sample should follow a normal distribution. We are trying to investigate the **sample mean which is coming from a normal distribution.** Another way to say the same thing is that, it assumes the sample mean should come from a normal distribution.

When we plot the sample means, it will follow a normal distribution (this is according to CLT) and for the sample which we had in hand at the very first place, we could say even that sample comes from the sampling distribution of sample means (distribution of sample means). We know that this distribution would be normal, so the sample mean in hand comes from a normal distribution and hence the normality assumption holds true.

That is why we had said that as the sample size increases, we are less concerned about the normality assumption as it is automatically taken care of by the Central Limit Theorem.

Why did we invest so much time in statistics?

Whatever we have covered so far in this book is important for validating assumptions and also important to get an idea about feature importance.

It is very important to make assumptions, the assumptions we make might be wrong, but we could use statistical tests to validate our assumptions.

Sometimes, even wrong assumptions could lead us to the right path.

Example - Suppose we have a Gender feature which contains 2 values Male and Female, we need to check if this feature has any impact on the target (continuous variable). We could run a t-test to check if Gender being Male or Female has an impact on the target column. If their means are not significantly different, we could possibly say that the Gender column with Male and Female values doesn't have an impact on the target (we need to take variance into account as well, also need to check for the assumptions of t-test before applying it).

But this is just to give you an idea about how we could go ahead building and validating assumptions using statistical tests.

Bonus content

Feel free to get access to some of the mini Ebooks we have created. You can get it at https://github.com/Dataebook/Ebook

Kaggle Exploration

This chapter would deal with the exploration of Kaggle, why is Kaggle important, what are some of the challenges people face, how to overcome those challenges and in the end, it would deal with a Kaggle problem statement. The entire code could be found on Github[1].

Why Kaggle?

There are many ways to learn and practice Data Science, then why does Kaggle hold a special place? Below are some of the questions we would like to ask.

Are you new to learning Data Science?

Are you the person who likes to learn Data Science through the application?

Assumption making is an important skill, do you want to broaden your assumption making skill?

Would you want to get a community and corporate support while you're learning to solve a Kaggle project?

Would you like to participate in competition post-learning or while learning Data Science?

If these are some of the questions you have in mind then the best solution we would suggest is *Kaggle*[2] - a platform only related to machine learning, data science, deep learning or AI stuff. **Kaggle** is a great place to try out something you newly learned and it will be very beneficial to you. **Kaggle** is not about competing with others in competitions rather it's a great platform to learn from the work other people have done.

Points to remember while exploring Kaggle:

- Don't panic when yo hear the term 'competition' when you are learning.
- Kaggle is more about learning.

[1] Github link - https://github.com/Dataebook/KaggleExploration

[2] Kaggle link - https://www.kaggle.com/

Mistakes made while exploring Kaggle

1. **Just looking and not experimenting -** Just having a look at how a kernel has solved a problem is not going to help any learner to learn anything. One has to take the code and execute it on a local machine for better understanding. Data science is an applied field, and the best way to solidify skills is by practicing and not looking.
2. **Failing to make prior assumptions -** Without making prior assumptions and validating them, you won't be in a place to make correct decisions, try to explore the data as much as possible and #FeelLikeABoss
3. **Failing to understand the kernel's point of view -** Whenever a person is trying to understand the solution of a kernel, he/she should try to keep ourselves at the place of the kernel contributor and should try to understand, why has that person made a certain set of assumptions, what might be his/her thinking process behind the solution they have made. Try to keep yourself under the shoes of the kernel contributor.
4. **Sticking to one Kernel's solution -** You should not restrict yourself to one kernel, try to explore as many kernels as possible, try to understand how the sample problem has been solved by different people, you would find something to learn from each one of them.
5. **Spending too much time on theory -** Many beginners fall into the trap of spending too much time on theory, whether it be math related (linear algebra, statistics, etc.) or machine learning related (algorithms, derivations, etc.). This approach is inefficient for 3 main reasons:
   - First, it's slow and daunting.
   - Second, you won't retain the concepts as well.
   - Finally, there's a greater risk that you'll become demotivated and give up if you don't see how your learning connects to the real world.
6. **Fail start using algorithms without knowing the math behind it -** Blindly using the algorithms hurts when a certain algorithm is not working as you wish and you cannot find the exact reason for that because you don't know the intuition behind the algorithm.
7. **Using the same algorithm for all the datasets -** For example, Random Forest or XGBoost is good, it works very well but, it is important to check other models as well.

How to overcome these mistakes?

1. **Change the way of approaching a problem statement -** Approach a problem statement by writing your assumptions. This makes one understand the domain very well. Make sure you make at least 5-10 assumptions before getting data in hand.
2. **Research why ML/DL/AI is required -** Every learner should be able to find out how machine learning can help any industry (For eg, the Banking sector) to address their problem. One can examine that by reading blogs or articles to find why machine learning or deep learning is needed to solve that particular problem.

   For example, let's say you are participating in a hackathon to predict the loan amount using machine learning and here you can observe that the problem statement is pretty much clear and you will start working with the data set directly without knowing the business case. For sure you will face difficulty while dealing with that particular hackathon.

   First, do think why machine learning is needed & research about how machine learning is being used in the banking sector while predicting the loan amount. Understand the business problem, which may in turn give you insights about **feature engineering skills** and you can observe changes in youself while you approach in this manner.

3. **Learn to be comfortable with partial knowledge -** You'll naturally fill in the gaps as you progress.
4. **Understand the Strength and Weakness of an algorithm -** It is always suggested to know the landscape of modern machine learning algorithms and their strengths and weaknesses.
5. **Practice:** It is very important to constantly practice problem-solving.

Challenges faced while dealing with Kaggle kernels

1. It would be quite challenging to understand the Kernel’s approach while learning.
2. The code might be new to the fresher which has been used by the kernel.

   For example, maybe we could have learned matplotlib and seaborn for visualization but some kernels would have used **Plotly** for visualization. Understanding that would be a bit difficult.

   Note: Remember that these are the common challenges faced by every learner. Every master was once a learner.

   “Every Olympic diver needed to learn how to swim first, and so should you.”

Our Experience

At the beginning, as soon as we read the problem statement, we would take the dataset in hand and try to understand the given features. But later we realized it is always better to understand the domain first before handling the dataset. We started making some assumptions and it helped us to get more acquainted with the data. Assumption making also helps in developing storytelling skills. It has helped us and would definitely work for you as well, give it a try.

While learning, we thought EDA is only about filling null values and putting it into Charts. But later we realized EDA is much more beyond those fancy visualizations. ***"EDA is a process of approaching the problem statement before Model building".***

There was a Kaggle project which we weren't able to understand and we got stuck in between to understand the code completely. And we spent many days understanding it. However, we found the best way to understand the code as well as the approach used in the kernel. It happened by taking ourselves at the position of Kernel contributor and seeing things from his/her perspective. If we are not understanding the code, we would try to understand the approach and store it in our mind. At any point in time, if a similar kind of approach comes, we would take the help of that code and manipulate it according to our requirement to complete our work. Because ***Data Science is not only about coding, but it’s more about giving solutions to the problem*** which would impact people's lives.

Now, let’s take one Kaggle problem and try to understand the EDA step by step. The focus of the chapter is not to build a model, rather tell you about how to approach a problem statement.

Note – We won’t be explaining the entire code in this book as it isn’t practically possible, we would just be covering the important code snippets. For the entire code, refer to the Github[1] repo.

How to deal with a problem statement?

Step1: Identifying Problem Statement

Many social programs have a hard time ensuring that the right people are given enough aid. It's tricky when a program focuses on the poorest segment of the population. This segment of the population can't provide the necessary income and expense records to prove that they qualify.

In Latin America, a popular method called "Proxy Means Test" (PMT) uses an algorithm to verify Income Qualification. With PMT, agencies use a model that considers the family's observable household attributes like the material of the walls and ceiling or the assets in their homes to classify them and predict their level of need.

While this is an improvement, accuracy remains a problem as the region's population grows and poverty declines.

The Inter-American Development Bank also believes that new methods beyond traditional econometrics, based on a dataset of Costa Rican household characteristics, might help improve PMT's performance.

Following actions should be performed:

1. Identify the output variable.

2. Understand the type of data.

3. Check if there are any biases in your dataset.

[1] Github link - https://github.com/Dataebook/KaggleExploration

4. Check whether all members of the house have the same poverty level.

5. Check if there is a house without a family head.

6. Set the poverty level of the members and the head of the house within a family.

7. Count how many null values are existing in columns.

8. Remove the null value from the target variable.

9. Predict the accuracy using a random forest classifier.

10. Check the accuracy using the Random forest with cross-validation.

Note: After reading the Problem Statement itself, we should make some assumptions without looking at the data.

Assumptions:

The given problem statement focuses on the social-economic side. We can solve any kind of problem from any domain with the help of enough data.

Let's start making our assumptions one by one for our problem statement before looking into the dataset:

Since the problem statement mentions household observable attributes, let's list down some of the attributes in our house and observable things. We have listed a few, based on our thought process.

- Refrigerator
- Mobile phones
- Laptop
- Desktop
- Water facility
- Toilet facility
- Wall type
- Ceiling type
- Fan, Light
- Electricity
- Stove
- Cylinder
- Kitchen room
- No. of rooms/house

- Type of Door lock
- Number of persons living per house
- Number of persons working/house
- Number of persons dependent on others/house - like children, grandma, grandpa(Age<18 and Age>65)
- Own house or rented house

Some of the observable attributes are mentioned above. You can list others from your end in case if we have missed any.

Let's start making our assumptions.

1. If a household holds all the above-mentioned attributes then definitely they should not be in the extreme poverty line.
2. Though a household holds all the above-mentioned attributes, if the dependency rate is high, maybe we can categorize them in a moderate poverty line.
3. If the wall is in bad condition, then it simply means that the household is not having enough money to renovate that. With this assumption, we can say that the household should be extremely poor. But it also depends on whether the house is owned or rented and other similar factors like that.
4. If there is no power supply for a household, for a long time, we can say that they do not have money to pay and get an EB connection for their home and so even they are in the extreme poverty line.
5. If in a house there are like 5members and if only one person is working and if the walls, ceilings, and floor material and if the home is rented, then he should be in the extreme poverty line.
6. If a house has only one room and no space for the kitchen, then definitely he should not be having good sanitary facilities. This kind of household should be focused on our program.
7. If the number of rooms is more than 3, then we can say that this household must own his house or pay his rent regularly. We can ignore this household as he is not in an extreme poverty line.
8. If a house doesn't have children and pregnant ladies, these people can manage themselves somehow. But if only one person is earning and considering he is living in a rented room and he has children and pregnant women, it becomes difficult for him to manage everything. So depending on his income, we can categorize him and target that household.

9. If there is no toilet facility for a household, it means that this particular household should be in the extreme poverty line.
10. If there is no one working in a house, and if more persons are living in that house, their daily food becomes a huge challenge.

These are the few assumptions we have made before looking into the dataset. Now with these assumptions let's look into the dataset and move forward.

Step 2: Understanding the weightage of Problem Statement

Before we look into the dataset, it's the responsibility of every Data Scientist to go to the environment to understand the exact weightage of the Problem Statement. Today we all have everything at our fingertips. If we just Google for something, it's going to give the complete history. So, first, let's understand "What is Costa Rica all about? What is IDB? What is a Proxy Means Test? What is the focus of the majority of social programs to help Costa Rica?"

Let's understand it now.

Costa Rica -

Costa Rica is one of the wealthiest countries in Central America, but a slum in the capital of San Jose, known as Carpio. This place is even ignored by society since the district is synonymous with poverty, drugs, and violence. As per a report by the National Statistics and Census Institute (INEC) in 2015, more than 1.1 million Costa Ricans lived in poverty, According to data from the National Household Survey of the National Institute of Statistics and Census, the percentage of poor households in Costa Rica went from 20.5 to 20 percent between 2016 and 2017, and extreme poverty dropped from 6.3 percent to 5.7 percent.

IDB -

The Inter-American Development Bank (IDB) is a cooperative development bank founded in 1959 to accelerate the economic and social development of its Latin American and Caribbean member countries.

Proxy Means Test -

In any social program, it is always tough to make sure that the aid goes to the deserving. The poorest segment of the population usually is unable to provide the necessary income and expense records to prove that they qualify. This is where proxy means testing (PMT) comes into the picture. The key idea is to use observable characteristics of the household or the members to estimate their incomes or consumption when other income data (salary slips, tax returns) are unavailable or sketchy.

What do we have to do? Identify what the problem statement it is all about -

We need to identify the "level of income qualification" needed for the families in Latin America based on the *household level* and also have to maximize the Accuracy score across all the categories. A generalized solution would be helpful to IDB and other institutions working towards helping the economically weaker sections of the society.

This is a supervised multi-class classification machine learning problem:

Supervised*: Provided with the labels for the training data*

Multi-class classification: *Labels are discrete values with 4 classes*

Each row represents one individual and each column is a feature, either unique to the individual, or for the household of the individual. The training set has one additional column, Target, which represents the poverty level on a 1-4 scale and is the label for the competition. A value of 1 is the most extreme poverty.

Step3: Identifying data sources and data understanding

Importing necessary libraries -

Here we would import all the required libraries.

Importing Dataset -

Now, it’s time to take the dataset in hand (It is recommended to have atleast 5 prior assumptions on how the variables would be and some hypotheses like we mentioned above).

```
print("Train Dataset - Rows, Columns: ", train_data.shape)
print("Test Dataset  - Rows, Columns: ", test_data.shape)

# Output
# Train Dataset - Rows, Columns:  (9557, 143)
# Test Dataset - Rows, Columns:  (23856, 142)
```

We are provided with 2 datasets; one is a **train** and the other one is a **test**. So, whatever changes we are doing to the training dataset, everything has to be done to the Test dataset as well. For eg, let's say if you find any null value in the Train dataset, and you are treating it in the training dataset and forgot to do the same in the test dataset, then our model would definitely go for a toss. Do you agree? So, make sure that you treat the Train and Test dataset equally when you are provided with two datasets - one with a Target column and the other one without a Target column.

Understanding the features –

We have been provided with a Train and Test dataset which is almost the same except the presence of the `'Target'` variable in the test dataset. Now, let's understand all the features by looking at its data type from the Train dataset.

The explanations for all 143 columns can be found on Github[1], but a few to note are below:

- `Id`: a unique identifier for each individual (this feature can't be used to train our model, but can be used to identify a person).
- `idhogar`: a unique identifier for each household. This variable is not an important feature but will be used to group individuals by the household as all individuals in a household will have the same identifier.
- `parentesco1`: indicates if this person is the head of the household.
- `Target`: the label, which should be the same for all members in a household.

If we observe each row with `parentesco1` column, we would understand that every record is on the individual level, with each individual having unique features and also information about their household. To create a dataset for the task, we'll have to ***aggregate individual's data*** for each household. Moreover, we have to make predictions for every individual in the test set, but ***"only the heads of household are used in scoring",*** which means we want to predict poverty on a household basis.

[1] Github link - https://github.com/Dataebook/KaggleExploration

Let’s look at the different values target contains and their frequency

```
train_data['Target'].value_counts()

# Output
# 4    5996
# 2    1597
# 3    1209
# 1     755
# Name: Target, dtype: int64
```

Target values represent poverty levels as follows:

1 = extreme poverty

2 = moderate poverty

3 = vulnerable households

4 = non-vulnerable households

Here, the majority of us would ask how to match 1 to the Extreme Poverty line and 4 to Non-vulnerable households. That’s where domain knowledge comes into the picture. Since this is a training dataset, based on the household attributes, each row is already categorized in one of these four classes.

Now the next question arises, is this how we are going to train our model? Isn't this an Imbalanced dataset? Is this dataset biased?

Continue Reading...

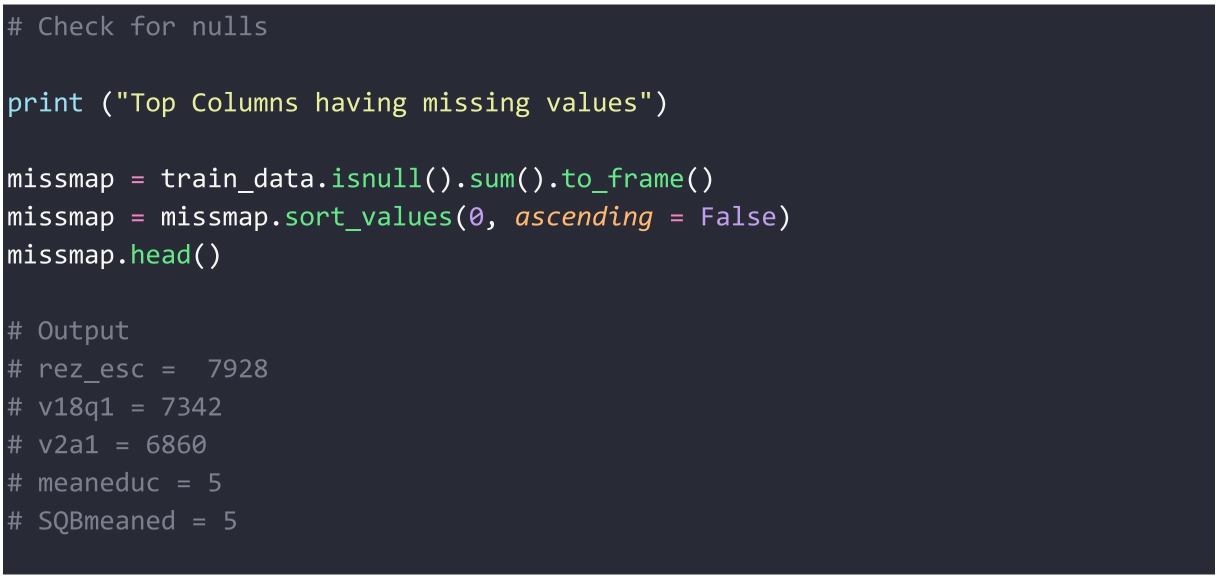
When we will make a model, we'll train on a household basis with the label for each household, i.e. the poverty level of the head of the household. The raw data contains a mix of both household and individual characteristics and for the individual data, we will have to find a way to aggregate this for each household. Some of the individuals belong to a household with no head of *the household* which means that unfortunately, we can't use this data for training. These issues with the data are similar to the ones we face in the **real world** and hence this problem is great for preparing you for the datasets you'll encounter in a data science job!

Step 4: Exploratory Data Analysis (EDA)

Every Data Scientist's most crucial role is EDA. If you have followed along till now, it would be very clear that EDA which is not meant *only for Null value treatment* and is far beyond that, it incorporates an important skill of Assumption making. We will go step-by-step on how to perform EDA for this particular problem statement.

Firstly, we have to check for null values in the dataset. If we find any, we have to treat it first based on domain knowledge before going any further and as we all know, ***Null value Treatment*** is only *one part of EDA* which one should start with.

Check for null values in the dataset –



```
# Check for nulls

print ("Top Columns having missing values")

missmap = train_data.isnull().sum().to_frame()
missmap = missmap.sort_values(0, ascending = False)
missmap.head()

# Output
# rez_esc =  7928
# v18q1 = 7342
# v2a1 = 6860
# meaneduc = 5
# SQBmeaned = 5
```

Here we can see 5 columns with null values in the entire dataset. Let's look at the data types and apply our common sense to fill the null values.

Fields with missing values -

`rez_esc` = Years behind in school

`v18q1` = Number of tablets household owns

`v2a1` = Monthly rent payment

`meaneduc` = Average years of education for adults (18+)

`SQBmeaned` = Square of the mean years of education of adults (>=18) in the household

Exploring the Null values –

When we explore the null values based on data type, we come to know that no nulls are present in columns with Integer and Object data type, the only nulls we have are for the float data type. And those are the same columns that we have listed above.

Before treating the null values, it is good to check if there are any mixed values in any column. It is known that if there are any mixed values, it should be in 'object data type', as we know Object data type can have numbers as well as characters.

Exploring object data types -

The object data types in the given dataset are `'Id'`, `'idhogar'`, `'dependency'`, `'edjefe'`, `'edjefa'`.

We can check for the unique values of the columns and then identify if the column is a mixture of alphabetical and numerical values.

By doing a check, we observe that there are 3 Object data types with mixed values. And they are `'dependency'`,`'edjefe'`,`'edjefa'`.

According to the documentation for these columns:

`dependency`: Calculated as the ratio of ‘number of members of the household younger than 19 or older than 64’ to ‘the number of members of the household between 19 and 64’.

`edjefe`: years of education of the male head of household, based on the interaction of `escolari` (years of education), `head of household` and `gender`, yes=1 and no=0

`edjefa`: years of education of the female head of household, based on the interaction of `escolari` (years of education), `head of household` and `gender`, yes=1 and no=0

For these three variables, it seems that “yes” = 1 and “no” = 0, we can change these variables by mapping them correctly.

Once we have ensured that no column is a mixture of alphabets and numbers, we can go ahead for Null value treatment. Remember that if mixed values are not changed then it is going to be a wrong approach.

Handling null values –

Null value treatment for `'rez_esc'` column -

Applying Common Sense:

`rez_esc` – (total nulls: 7928) = Years behind the school.

Anyone younger than 7 or older than 19 presumably has no years behind and therefore the value should be set to 0. For this variable, if the individual is over 19 and they have a missing value, or if they are younger than 7 and have a missing value we can set it to zero.

Null Value treatment for `'v18q1'` column -

Applying Common Sense:

`v18q1` – (total nulls: 7342) = Number of tablets household owns

In our dataset, we have a column `'v18q'` which indicates - whether or not a household owns a tablet. So, applying our common sense we can say that if a household owns a tablet, then the count would be some number in the ‘`v18q1`’ column, if not the count should be zero. So, we can replace Nan with zero by checking with ‘`v18q`’.

Null Value treatment for `'v2a1'` column -

Applying Common Sense:

`v2a1` – (total nulls: 6860) = Monthly Rent Payment

Why only look at the null values? Let's look at a few rows with nulls in `v2a1`.

Columns related to Monthly rent payment are listed down:

`tipovivi1`= 1 own and fully-paid house

`tipovivi2`= 1 own, paying in installments

`tipovivi3`= 1 rented

`tipovivi4`= 1 precarious

`tipovivi5`= 1 other (assigned, borrowed)

By using all these columns, it makes sense that when the house is fully paid, there will be no monthly rent payment. So, we can add 0 for all the null values in ‘v2a1’, when the house is fully paid.

Null Value treatment for 'meaneduc' column -

Applying Common Sense:

meaneduc – (total nulls: 5) = Average years of education for adults (18+)

Let's look at the columns related to average years of education for adults (18+):

- **edjefe** -> years of education of the male head of household, based on the interaction of escolari (years of education), head of household and gender, yes=1 and no=0.
- **edjefa** -> years of education of the female head of household, based on the interaction of escolari (years of education), head of household and gender, yes=1 and no=0.
- **Instlevel1** = 1 -> no level of education
- **Instlevel2** = 1 -> incomplete primary

```
data = train_data[train_data['meaneduc'].isnull()].head()
columns=['edjefe','edjefa','instlevel1','instlevel2']
data[columns][data[columns]['instlevel1']>0].describe()
```

By this, we can see that the meaneduc is null when no level of education is 0. So, we can replace all the Nan’s with 0 to fix the meanedu column.

The things won’t be crystal clear here, refer to Github link for the entire code with proper explanation.

Null Value treatment for ‘SQBmeaned’ column -

Applying Common Sense:

SQBmeaned – (total nulls: 5) = Square of the mean years of education of adults (>=18) in the household.

This column is just the square of the mean years of education of adults.

```
data = train_data[train_data['SQBmeaned'].isnull()].head()
columns=['edjefe','edjefa','instlevel1','instlevel2']
data[columns][data[columns]['instlevel1']>0].describe()
```

The above code tells us that the `SQBmeaned` is null when no level of education is 0. So, we can replace all the Nan's with 0 to fix the '`SQBmeaned`' column.

Almost all the null values have been treated properly with the domain and data understanding.

Note: When we observe, we can see that all the null values are replaced with zero in this problem statement. Do you think it is replaced just like that? No. Based on the domain knowledge and common sense, it got perfectly replaced. This is how Null value treatment has to be done.

Removing insignificant columns/features

Let's look at the Squared Variables -

`SQBescolari` = `escolari` squared

`SQBage` = `age` squared

`SQBhogar_total` = `hogar_total` squared

`SQBedjefe` = `edjefe` squared

`SQBhogar_nin` = `hogar_nin` squared

`SQBovercrowding` = `overcrowding` squared

`SQBdependency` = `dependency` squared

`SQBmeaned` = square of the mean years of education of adults (>=18) in the household.

When we look at the description document, we can see that the above-squared variables convey similar meaning as the non-squared variables already in the dataset. Anyone could go for model building. So we can remove them.

Column Definitions

As a part of the analysis, we have to define the columns that are at an individual level and a household level using the data descriptions. There is simply no other way to identify which variables are at the household level, other than going through the variables themselves in the data description.

We'll define different variables because we need to treat some of them differently. Once we have the variables defined on each level, we can start aggregating them as needed.

The process is as follows:

1. Break variables at a household level and an individual level.

2. Find suitable aggregations for the individual-level data.

- Ordinal variables can use statistical aggregations.
- Boolean variables can also be aggregated but with fewer stats.

3. Join the individual aggregations to the household level data.

Define Variable Categories -

There are several different categories of variables:

1. Id variables: Identifies the data and should not be used as features.

2. Individual Variables: These are characteristics of each individual rather than the household.

- Boolean: Yes or No (0 or 1).
- Ordered Discrete: Integers with an ordering.

3. Household variables

- Boolean: Yes or No
- Ordered Discrete: Integers with an ordering
- Continuous numeric

Removing Correlated features –

We can draw a correlation matrix to find the correlated features. We come to know that the below features have something to do with the size of the house.

- r4t3 = Total persons in the household
- tamhog = size of the household
- tamviv = number of persons living in the household
- hhsize = household size
- hogar_total = # of total individuals in the household

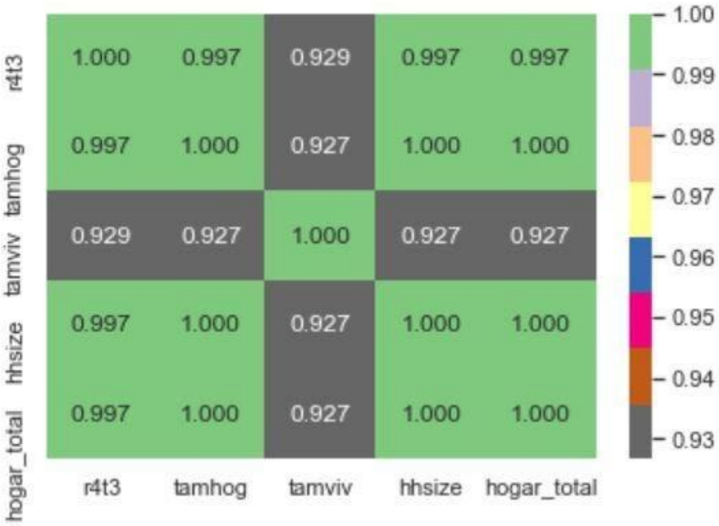


Figure 16: Correlation heatmap

With the help of the correlation score, the features which show the correlation score greater than 90% are identified and removed to avoid redundancy. Similarly, you have to work on finding the redundant variables in the dataset and try to remove them before building the model.

Note: Remember whatever changes you are doing to the train dataset, it has to be applied to the test dataset as well.

For complete code with detailed steps, refer to Github[1].

"EDA IS A PROCESS OF APPROACHING A PROBLEM STATEMENT BEFORE MODEL BUILDING".

[1] Github link - https://github.com/Dataebook/KaggleExploration

Churn Prediction

Brief about the problem statement

Churn prediction is all about retaining the customers. When you offer a service, there might be a group of people who are not likely interested in availing the service that is being offered by you. Such a set of customers are more likely to stop using your service in the coming time, these set of customers are referred to as churned customers.

This is not just true for services, this might also happen with products as well. Customers might start disassociating themselves from the product with which they had been associated since long. The reason for this might be any.

Customer churn happens when customers stop doing business with a company.

Did you ever think why service providers like AIRTEL and IDEA give different offers to different customers even when they have over millions of customers?

It might have happened that you might have got an offer for Rs 450 and your friend or a family member might have got a similar offer at a cheaper price from the same service provider.

Why does this happen??

Let's understand what churn is.

If you are switching from Airtel to Idea, it is known to be a churn for Airtel. If you switch then churn rate = 1 which means you have stopped using the services from your service provider, if not then churn rate = 0 which means you are continuing to use their service. But what sense would it make to identify a customer after he/she has churned so we want to prevent customer churn hence when a customer is about to churn we set churn rate = 1. So in such use cases when churn rate = 1 then the customer is churned or is about to churn. A business needs to identify such customers. The reason they give offers to some people at cheap prices is to retain their customers by giving attractive offers.

This is a use case where we could use Data Science to get to the solution. We can analyze from the above problem statement that to predict whether a particular customer is going to change the service provider or not becomes a major challenge to deal with.

In simple words, we have to predict customers who are most likely to discontinue our service in the future and focus on how to retain such customers.

Here we will have three major Business Challenges to deal with -

1. We need to focus on Prepaid Customers, the number of individuals using prepaid services is more and it is important for any service provider that they focus on such customers for continued revenue generation; it's an important and major factor to work with such customers.
2. We also need to focus on Post-paid Customers but in this case, the number of customers will be less, and also we cannot afford to lose such customers as they help the company for continued revenue. But it should be pretty clear that we use different strategies for Prepaid and Postpaid customers hence we see different offers for Prepaid and Postpaid customers.
3. Many service providers/companies give **FREE** offers for a small duration of time and the company assumes that they would attract customers and show them how good their company is when it comes to providing service to their customers. But many smart customers just consume these free services and keep hopping in search of such freebies and this is called **Fraud Activation**. This hampers the conversion ratio of the company as they fail to convert potential customers to actual customers, so companies need to deal with such scenarios as well. If companies fail to convert customers after offering freebies this leads to a huge loss to the company.

Churn Prediction is one of the common business use case across all the industries, be it telecommunication or any product based company.

But as a Data Scientist, our job is not to always build ML models with 92% or 95% accuracy, rather our focus is to build models that are interpretable and can help us derive some insights. In real life, our goal is to keep things simple and not to go for complex models until required, as complex models are difficult to interpret. And even if you could interpret that model, not everyone would be as smart as you, you aren’t the only person working on the project. Also, you won’t be marrying the organization to stay with the organization for a lifetime. So there is a trade-off between model complexity and model interpretability, we are just aware of the Bias Variance trade-off but model complexity and model interpretability trade-off is also important when working on real-life projects.

Why is this problem statement important?

Many of us don’t understand the problem statement and directly jump to the dataset aiming at building a model with 95% accuracy, but we fail to understand whether our model is learning or memorizing.

Building assumptions and use of common sense plays a major role in the Data Science domain, so we need to use our common sense to come up with assumptions and statistical tests would help us to validate those assumptions.

Before diving into the data, it is important to understand the importance of the problem. What change would you bring in after solving this problem? if the change isn’t that big, it won’t motivate you to solve the problem. That is why it is important to understand the weightage of the business problem that you are going to solve.

Interact with your client, ask him how your work is going to affect their organization and trust me, the client would be happy to see that you are interested in understanding their business and not just working like a bot. And the importance of the business problem would keep you motivated throughout the process as there might be many instances where you might get demotivated because of the challenges you might face during the entire process.

Retention of a customer is cheap as compared to getting new customers. The old customers build a certain level of trust in their minds for the company when they associate with a company for some time duration. But when it comes to getting new customers, it is a costly job because money is spent on drawing the attention of new customers towards your company.

New customers are hesitant when it comes to establishing a new relationship. It is but obvious that even we might be hesitant in trying a new product. So to deal with this, the new company might offer some services for free and after that, there might be chances that the new customer could potentially get engaged with the company. But this isn’t that easy, out of around 1000 customers only a handful might turn into leads. The conversion from Potential customers to a Lead is quite challenging, the conversion ratio is quite low.

This is what **Michael Redbord**, general manager of Service Hub at **HubSpot** has to tell about churn prediction:

"In a subscription-based business, even a small rate of monthly/quarterly churn will compound quickly over time. Just 1 percent monthly churn translates to almost 12 percent yearly churn. Given that it's far more expensive to acquire a new customer than to retain an existing one, businesses with high churn rates will quickly find themselves in a financial hole as they have to devote more and more resources to new customer acquisition."

Gain knowledge about business/domain.

It is but obvious that different companies might belong to different domains and so the data they collect will be different. For example, the telecom industry might collect data related to the number of incoming calls, the duration of the call, etc.

Depending on the domain, the data captured varies, and depending on the data captured we might target different features. In such a scenario business knowledge becomes of utmost importance. For example, consider a customer who was quite frequent when it came to calling and now for around 20 days if there are no outgoing calls from that customer then there are chances of that customer getting churn.

There might be many other features that are important but we will get an idea about that only after we gain ample knowledge about the domain or about how the business works.

Price also plays an important role, it is observed that high priced subscriptions are less prone to churn as compared to low priced subscriptions because high price subscriptions are more considered (people think twice before subscribing to something costly). Subscribers sign up and cancel more often when the price is low.

If you think of buying a subscription you would be more cautious about buying it if it's expensive and you would buy it only if you are sure that you would use that particular service. If it's available at a cheap price you won't think twice before buying it, as investing a lesser amount doesn't matter much.

Handling Missing Values

As we know, handling missing values is one of the major skill and most individuals fail to understand its importance.

Most of us just delete Null Values if the percentage is about 40% or more than 40% but it depends on the importance of that feature with regards to our problem statement.

If we take one small example from the telecommunication industry (churn prediction), where we have one of the features named Total_Number_of_Inactive_days, it tells us since how long is the customer inactive or didn’t use any of our services (like calling, data, messaging, etc). Using this feature we can draw some insight that if this feature has a higher value then the customer might churn.

Let’s say that we have around 30% or 40% null values, in this case, we might fill the Null values using forward fill, backward fill, mean, median or mode. But how would we be sure that the imputation technique which we have applied to fill the Null values is correct?

Consider the feature Total_Number_of_Inactive_days, it might happen that by using forward or backward fill we might impute incorrect values leading to incorrect decisions. Suppose that we have a null value for this feature and the value above this row is 0, indicating that the customer is active and not likely to churn but just because we have used forward/backward fill we might impute the null value with 0 indicating that there is no chance for that customer to churn. But it might happen that the customer might be on the way to churn. So we should handle null values wisely by thinking about all the possible scenarios.

Before imputing missing values, we should check if the imputation makes sense to the business, for this, we should have ample business knowledge to xome to a final conclusion.

Approach followed by leading companies.

There are many techniques followed by leading companies to deal with the issue of Customer Churn but below are one's we found quite interesting.

Customer Segmentation -

In this, the customers are segmented into groups depending on the type of services they avail, their needs, level of engagement, the monetary value, their feedback, and a lot of many things. These customer categories share a common belief and common behavior patterns and this allows us to focus on these groups/categories rather than focussing on the entire set of customers. If we try to focus on the entire set of customers, it might happen that different customers might have faced different issues but some people might have similar issues so they could be grouped and targeted collectively. So instead of building a model using the entire set of customers, we can build models that are specific to a group/category representing the entire segment. This method is observed to result in a far better solution than targeting all the customers in a similar way.

Focus on the outcome -

Another technique that is followed by many leading companies is that, they suggest just tracking the user data isn't sufficient, user data includes how often the user interacts with the website, how is his/her engagement, how often the user completes the given assignment, etc. It is equally important to track the outcome (result) the user is getting out of the product or service he/she has availed. It is important to understand the results because if the users aren't getting any value out of the product/service then they are likely to churn.

For example, if your company launches a course and you see that the users are interacting quite often with your website also they are submitting the assignments on time. But if they are unable to get any value out of that course then it's most likely that they won't join any other course provided by your company and also they won't suggest your course to any of their connections. So, in this case, we are not only losing our existing customers but we are also closing the doors for our new customers.

Observation window -

We aim to get some idea about the behavior of the customers who have churned and then check if our existing customers are following the same pattern. If we get to see a similar pattern then this means that a specific customer might churn. So we observe the behavior of customers for a particular period (called a window), this ends before a specific point in time and we make predictions about a period or window that starts after the **observation window** has ended. The observation window is the customer history which contains how the customer behaved in the past and the prediction window is also called a **performance window**, it is the one where we try to predict if the customer will churn depending on the customer history.

Spotify had incorporated a similar approach when they were new in the market. They had offered free membership for a month and they checked if the people were using their app in the second week after the registration. If they see that the customers are not using it in the second week then the customers are likely to churn. So they had 3rd and 4th week in hand (called re-engagement week) where they tried to re-engage the customers who had stopped using services after the 2nd week. In their case, the observation window was 2 weeks because they had observed that in the 2nd-week customers are most likely to churn. This was specific to their business, and it varies with business.

So the correct knowledge of the size of the window comes with business knowledge and experience. Too short observation window might give you ample time for re-engagement but you might not be able to correctly identify the behavior of the customer.

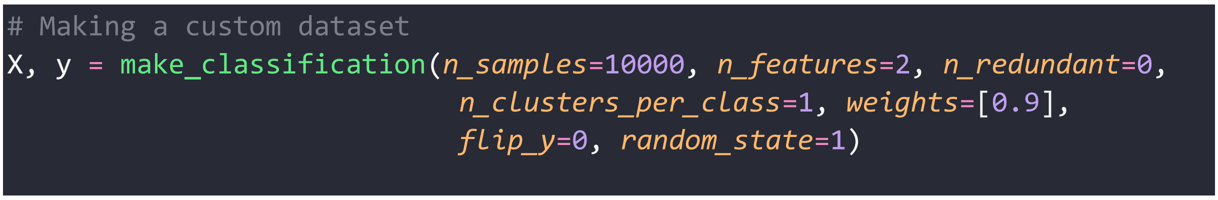
Dealing with imbalanced Dataset.

This is one of the challenges we face when dealing with the Churn Prediction use-case. What do you mean by an imbalanced data set?

For example, assume that you are working on a Fraud detection project so let’s say that there are 1000 records, it is evident that around 10-20% people might be fraud, so observations belonging to fraudulent class would be 10-20% and similarly, non-fraudulent observations would be 80-90%. In such scenarios, before proceeding to model building it is important to handle the imbalance in data. Why is it important? that is something which we will cover in the next chapter. We would look at different techniques to handle an imbalance in the next chapter.

Imbalance classification

We will create a custom dataset to see how an imbalance in data affects the process.



```
# Making a custom dataset
X, y = make_classification(n_samples=10000, n_features=2, n_redundant=0,
                           n_clusters_per_class=1, weights=[0.9],
                           flip_y=0, random_state=1)
```

Our data has around 90% samples for class 0 and 10% samples for class 1.

From the below plot we can see that our data has a huge imbalance

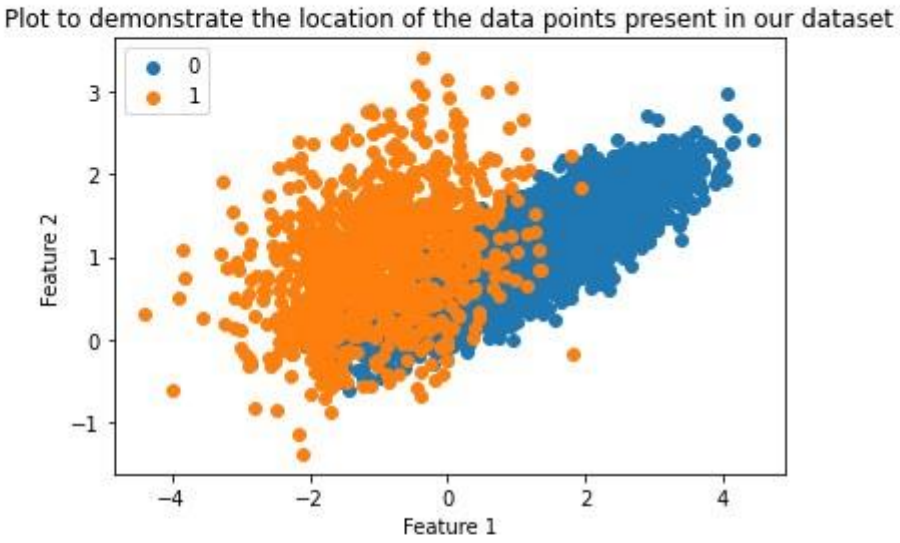
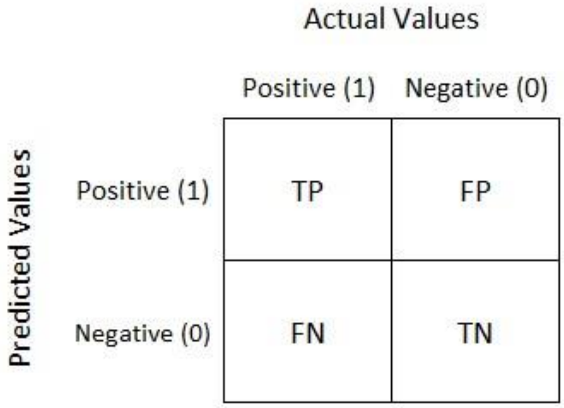


Figure 17: Imbalanced data

Before we proceed, let us try to understand some terminologies related to Classification metrics.

1. **Confusion matrix** - It depicts how our model is performing by plotting the predicted labels VS the actual labels.
   True Positive (TP) - This means that the prediction of our model is True (correct) and the prediction made by model is 1 (+ve).
   True Negative (TN) - This means that the prediction of our model is True (correct) and the prediction made by model is 0 (-ve).
   False Positive (FP) - This means that the prediction of our model is False (incorrect) and the prediction made by model is 1 (+ve).
   False Negative (FN) - This means that the prediction of our model is False (incorrect) and the prediction made by model is 0 (-ve).



		Actual Values	
		Positive (1)	Negative (0)
Predicted Values	Positive (1)	TP	FP
	Negative (0)	FN	TN

This is the base for all the metrics that we might come across.

2. **Sensitivity** - Also called as Recall and True Positive Rate. It tells us how well our models predict the positive labels out of all the positive labels.

$$\mathit{Sensitivity}\ (\mathit{Recall}) = \frac{\mathit{True\ Positives}}{\mathit{True\ Positives}+\mathit{False\ Negatives}}$$

3. **Specificity** - It tells us how well our model predicts the negative labels out of all the negative labels, also it is the fraction of the total amount of relevant instances that were actually retrieved.

$$\mathit{Specificity} = \frac{\mathit{True\ Negatives}}{\mathit{True\ Negatives}+\mathit{False\ Positives}}$$

4. **False Positive Rate (FPR)** - It is the ratio of labels that were incorrectly classified as positive (when they are actually negative) to the total number of negative labels.

$$\mathit{FPR}\ (1-\mathit{Specificity}) = \frac{\mathit{False\ Positives}}{\mathit{True\ Negatives}+\mathit{False\ Positives}}$$

5. **Precision** - It is the ratio of correctly predicted positive labels to all the labels predicted as positive by our model (correct + incorrect). It tells us how correctly our model has predicted the positive labels out all the labels that our model has predicted as positive. Also, it is the fraction of relevant instances among the retrieved instances.

$$\mathit{Precision}\ = \frac{\mathit{True\ Positives}}{\mathit{True\ Positives}+\mathit{False\ Positives}}$$

Let us understand Precision and Recall with a real-life example –

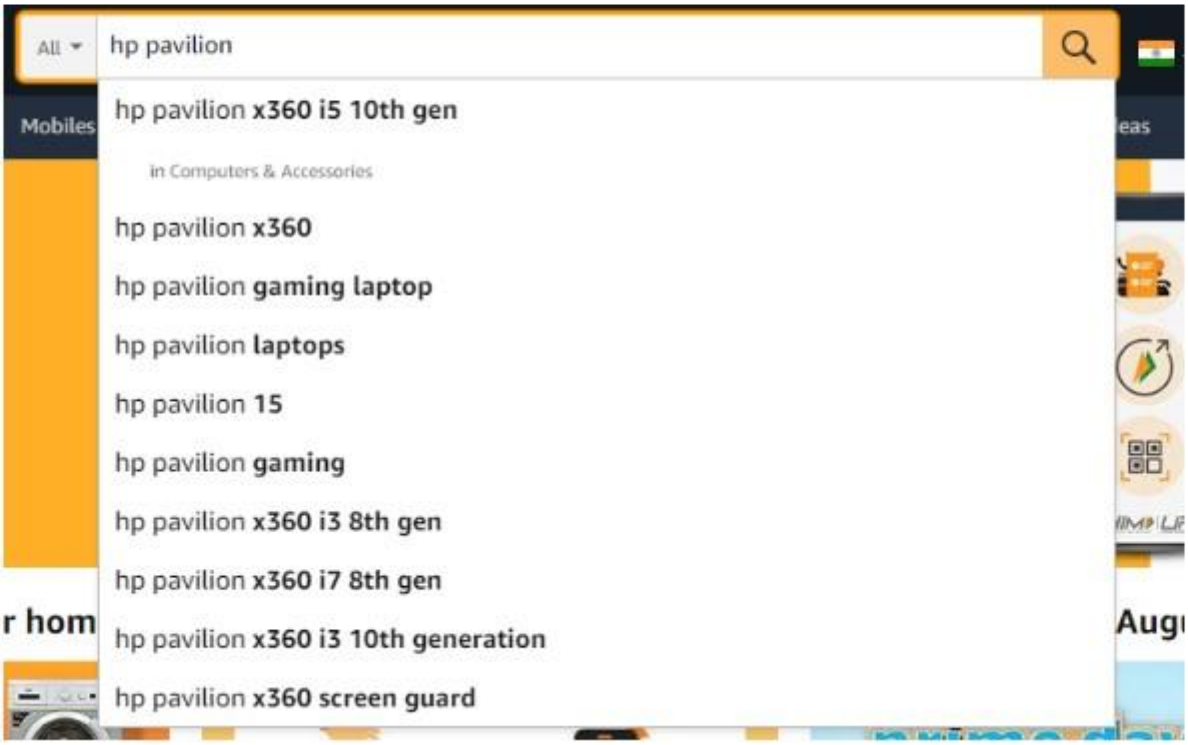

All
hp pavilion
Mobiles
hp pavilion x360 i5 10th gen
in Computers & Accessories
hp pavilion x360
hp pavilion gaming laptop
hp pavilion laptops
hp pavilion 15
hp pavilion gaming
hp pavilion x360 i3 8th gen
hp pavilion x360 i7 8th gen
hp pavilion x360 i3 10th generation
hp pavilion x360 screen guard

Figure 18: Organic search

In the image above we have searched for 'hp pavilion', *so the ratio of relevant suggestions out of all the rendered suggestions becomes our Precision*.

Recall would be the ratio of the relevant suggestions rendered to all the relevant suggestions that exist.

The suggestions we could see aren't the only relevant suggestions, there are many other terms/strings which are relevant to the search query.

So we should thoroughly understand the problem statement before deciding which metric to use.

Suppose we build an AI system to list the corrupt people, in this case, we would be interested in High Recall as we would want to identify all the corrupt people that exist as missing anyone of them is injustice and harm to society.

If correctly identifying the positives is our goal then we should go with Sensitivity (Recall) but if our goal is to correctly identify the negative then we should go for Specificity. So the choice of metric to use completely depends on what you aim to achieve.

We should not use Accuracy when the data is imbalanced. Why?

Suppose in the above data set with features `X` and target `y`, if we build a model which always predicts the majority class let us see what will happen.

We will fill `y_pred` with the mode of the target column i.e. `y`

```
# Populate y_pred with the value which has max freq.
y_pred = np.full(shape=y.shape, fill_value=stats.mode(y)[0][0])
accuracy_score(y, y_pred)
# Output - 0.9001
```

This gives us an `accuracy` of 90% as we have classified all the samples as the majority class, but we have badly failed to classify any of the samples of the minority class.

```
# Print Precision and Recall
print('Precision: {:.2f}, Recall: {:.2f}'.format(precision_score(y, y_pred),
                                                   recall_score(y, y_pred)))
# Output - Precision: 0.00, Recall: 0.00
```

Let us try re-creating a similar dataset but with a twist.

Earlier we had `weights=[0.9]` and now we have `weights=[0.1]`. It is doing the same thing, imbalance remains the same.

Now we represent minority class with `label 0` and majority class with `label 1` but earlier we had represented minority class with `label 1` and majority class with `label 0`.

```
# Value of weights parameter changed
X1, y1 = make_classification(n_samples=10000, n_features=2,
                             n_redundant=0, weights=[0.1],
                             n_clusters_per_class=1, flip_y=0
```

Our data has around 90% samples for class 1 and 10% samples for class 0.

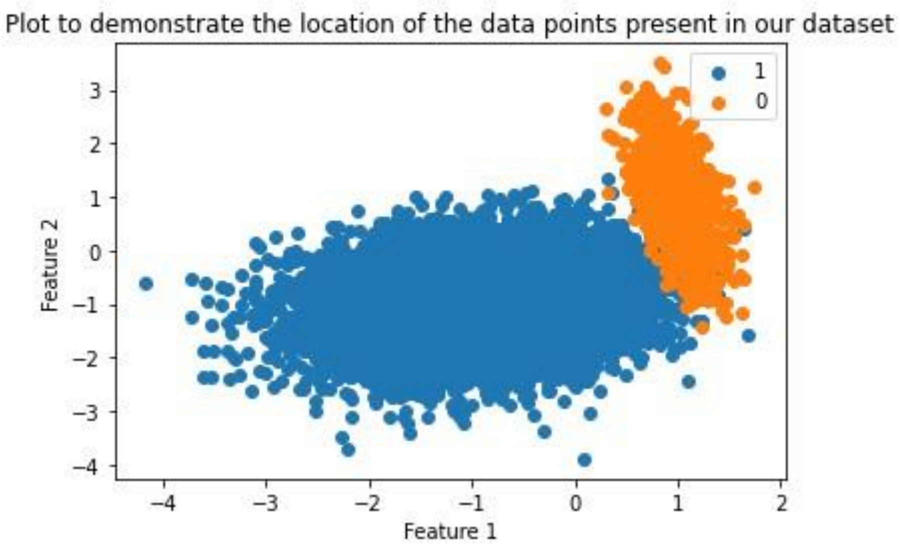


Figure 19: Imbalanced data

We can see that earlier `label 1` was our minority class which has now become our majority class. Let us do the same that we have done above and then check the metrics.

```
# Doing same as we have done above
y_pred_new = np.full(shape=y_new.shape, fill_value=stats.mode(y_new)[0][0])
accuracy_score(y_new, y_pred_new)
# Output - 0.9
```

```
# Print Precision and Recall
print('Precision: {:.2f}, Recall: {:.2f}'.format(precision_score(y_new,
                                                y_pred_new), recall_score
                                                (y_new, y_pred_new)))
# Output - Precision: 0.90, Recall: 1.00
```

The accuracy is still the same, but the `Precision` and `Recall` have improved terribly. High `Precision` and High `Recall` is what we strive for. This means that our model is fabulous. And it is performing awesome even without using **Machine Learning.**

But we know that our model would badly fail to correctly predict the minority class. So our belief that `Precision` and `Recall` could help us when we have imbalance data, failed.

This doesn't mean that `Precision` and `Recall` don't help when we deal with imbalanced data, they are super useful, but `Precision` and `Recall` have a huge hatred towards 0 and this is the case with most of the stuff in Data Science. This is what we have been taught, zero is bad, so we tend to ignore zero, same is the case with `Precision` and `Recall`.

Had it been the case where `label 0` was the minority class and `label 1` was the majority class, Precision and Recall would have told us that our model is poor. We have already tried this at the start.

Usually, we tend to see a severe imbalance in Fraud Detection, Churn Prediction, etc. In such cases our focus is to correctly predict the minority class, so we want to give more importance to the minority class. If such is a case, then we should represent the minority class using `label 1`, if we do so then we could trust the result given by `Precision` and `Recall`.

Generally, it is suggested to label the minority class with `label 1` so that we could trust `Precision` and `Recall`.

The same would be the case with the `F1 score` because the `F1 score` uses `Precision` and `Recall` for its calculation. And also when `Precision` and `Recall` is high `F1 score` is high.

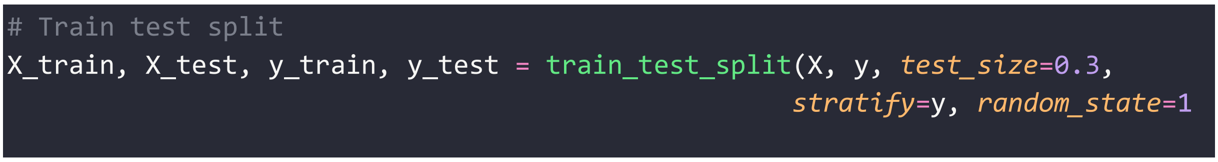
But an important thing to note is that it is very difficult to get a high `F1 score`, as a high `F1 score` would require High `Precision` and High `Recall` but it is difficult to achieve. Because when we try to achieve High `Recall` we tend to say that we want to predict all the positive labels correctly which also leads to increase in `False positive`, increase in `False positive` leads to decrease in `Precision` (this is also evident from the formula)

If we consider the Churn prediction use-case, here if 100 customers might leave our service and out of those we predict that just 20 are about to churn. Let's say out of those 20, 19 are correct (they would churn) so our `Precision` is quite high in this case. As `Precision` tries to check out of the 20 we predicted how many were correct, which is 19 in our case and that's quite good. But `Recall` tries to check out of 100 how many customers have we correctly identified, so that ratio is quite low as we have just identified 19 out of 100.

We aim to identify maximum customers correctly and so `Recall` makes sense, increasing `Recall` might increase `False Positive` but `False positive` is not a major concern in this use-case as even if we identify a customer who won't churn as a customer who will churn, there's not much harm in that, but if we fail to identify a customer who will churn in future as a customer who won’t churn, then that's a huge problem.

So the selection of a metric depends on what we want to achieve.

Now let’s try building a model the usual way i.e. using `train_test_split`.



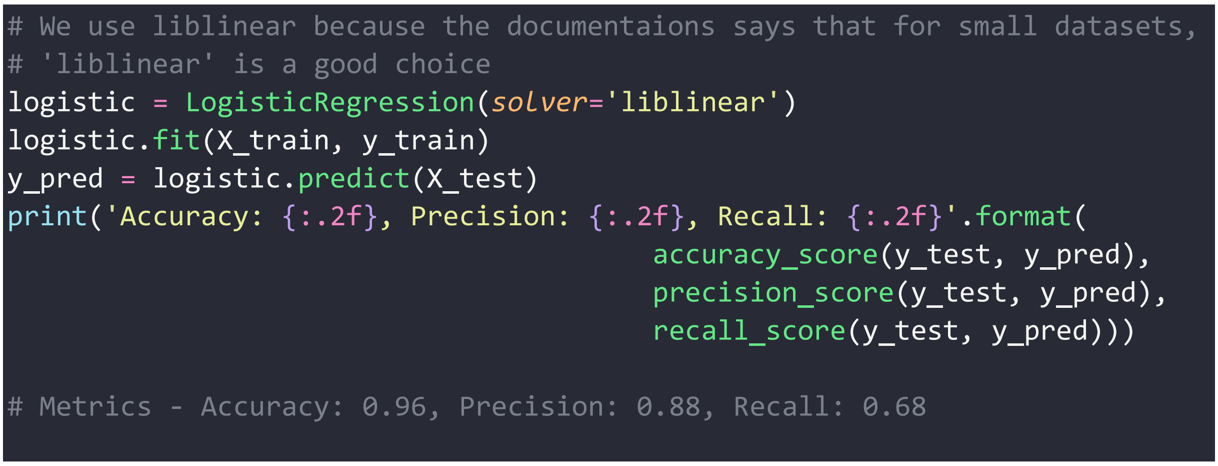
```
# Train test split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3,
                                                   stratify=y, random_state=1
```

The `y_train` and `y_test` have a distribution similar to the original dataset i.e. 90% of negative samples. This has happened because we have used the `stratify=y` parameter.

If we don't use `stratify=y` parameter, it might happen that the distribution won't be maintained and some classes would be present in train data and no record of a particular class might be there in test data

`stratify` parameter is used to demonstrate the importance of stratified sampling which we studied in the statistics section of this book.

To handle imbalanced data we would use `imbalanced-learn` package



```
# We use liblinear because the documentaions says that for small datasets,
# 'liblinear' is a good choice
logistic = LogisticRegression(solver='liblinear')
logistic.fit(X_train, y_train)
y_pred = logistic.predict(X_test)
print('Accuracy: {:.2f}, Precision: {:.2f}, Recall: {:.2f}'.format(
                                        accuracy_score(y_test, y_pred),
                                        precision_score(y_test, y_pred),
                                        recall_score(y_test, y_pred)))

# Metrics - Accuracy: 0.96, Precision: 0.88, Recall: 0.68
```

Undersampling

RandomUnderSampler

Under-sample the majority class(es) by randomly picking samples with or without replacement.

```
# Using RandomUnderSampling
undersampled_data = RandomUnderSampler(sampling_strategy=0.5)
X_under, y_under = undersampled_data.fit_resample(X_train, y_train)
```

`sampling_strategy` parameter specifies the sampling information, a `float` is only used for binary classification, it throws an error for multi-class classification.

If we set `sampling_strategy = 0.1` that means the ratio of the number of observations of minority class to the number of observations of the majority class would be 0.1

```
# We use liblinear because the documentaions says that for small datasets,
#  'liblinear' is a good choice
logistic = LogisticRegression(solver='liblinear')
logistic.fit(X_under, y_under)
y_pred = logistic.predict(X_test)
print('Accuracy: {:.2f}, Precision: {:.2f}, Recall: {:.2f}'.format(
                                      accuracy_score(y_test, y_pred),
                                      precision_score(y_test, y_pred),
                                      recall_score(y_test, y_pred)))

# Metrics - Accuracy: 0.94, Precision: 0.66, Recall: 0.82
```

Oversampling

RandomOverSampler

Over-sample the minority class(es) by picking samples at random with replacement.

```
# Using Random Over sampling
oversampled_data = RandomOverSampler(sampling_strategy=0.5)
X_over, y_over = oversampled_data.fit_resample(X_train, y_train)

# Metrics - Accuracy: 0.94, Precision: 0.64, Recall: 0.83
```

SMOTE

SMOTE(Synthetic Minority Over-sampling Technique) works by selecting examples that are close in the feature space, drawing a line between the examples in the feature space, and drawing a new sample as a point along that line.

```
# Using SMOTE for Over sampling
oversampled_data = SMOTE(sampling_strategy=0.5)
X_smote, y_smote = oversampled_data.fit_resample(X, y)

# Metrics - Accuracy: 0.94, Precision: 0.64, Recall: 0.83
```

Combining Undersampling and Oversampling.

We could either combine Undersampling and Oversampling manually or we could use inbuilt functionalities to help us achieve this.

Combining them manually -

We would first apply oversampling with a sampling_strategy=0.2 so this would create X and y where the ratio of minority class to majority class would be 0.2. This means that the number of observations in minority class would be 20% of the majority class.

So majority class = 6301 observations and minority class = 1260 observations.

After this we would apply undersampling to under-sample the majority class, here sampling_strategy=0.5 would create X and y where the ratio of minority class to majority class would be 0.5

So majority class = 2520 observations and minority class = 1260 observations, also minority to majority ratio is 0.5.

First, we apply over-sampling -

```
over_sampler = RandomOverSampler(sampling_strategy=0.2)
under_sampler = RandomUnderSampler(sampling_strategy=0.5)
X_over, y_over = over_sampler.fit_resample(X_train, y_train)
minority, majority = Counter(y_over)[1], Counter(y_over)[0]
print('{:.1f}'.format(minority/majority))
# Output - 0.2
```

Then we apply under-sampling on the over-sampled data -

```
X_under, y_under = under_sampler.fit_resample(X_over, y_over)
minority, majority = Counter(y_under)[1], Counter(y_under)[0]

# Here minority/majority => 0.5
# Metrics - Accuracy: 0.94, Precision: 0.65, Recall: 0.82
```

After building a model, we get above metrics, in our case the metrics are the same as we only have 2 features but you would observe a difference in metrics when you apply these techniques on datasets with more number of features.

Pipeline

There are many scenarios where we have a fixed set of transformations to be performed on the data, post which we would like to fit a model. All these things need not be done separately, we can create a pipeline and add all the transformations followed by the model we want to fit into the pipeline. Using the pipeline makes our life easier.

Intermediate steps of the pipeline must be 'transformations', that is, they must implement `fit` and `transform` methods. The final estimator only needs to implement `fit`.

Below we create a pipeline where we first add `oversampler` followed by `undersampler` which is then followed by a ML model which we would like to fit.

```
# Initilaize pipeline with the required steps
pipeline = Pipeline([('smote', oversampled_data),
                     ('under', under_sampler),
                    ('model', LogisticRegression())])
scoring = ['accuracy', 'precision', 'recall']
cv = RepeatedStratifiedKFold(n_splits=10, n_repeats=3, random_state=1)
# evaluating the model
scores = cross_validate(pipeline, X, y, scoring=scoring, cv=cv, n_jobs=-1,
                        return_train_score=True)
print('Accuracy: {:.2f}, Precision: {:.2f}, Recall: {:.2f}'.format(
                                   np.mean(scores['test_accuracy']),
                                   np.mean(scores['test_precision']),
                                   np.mean(scores['test_recall'])))

# Metrics - Accuracy: 0.91, Precision: 0.52, Recall: 0.88
```

Here we can observe that `recall` is `0.88` and we have used Cross-Validation so the model is tested on entire data. So this result is always better than the result that we obtain from `train_test_split` as it is more trustful.

Combining SMOTE oversampling with any other undersampling method is found to perform better.

Pre-defined Oversampling with Undersampling

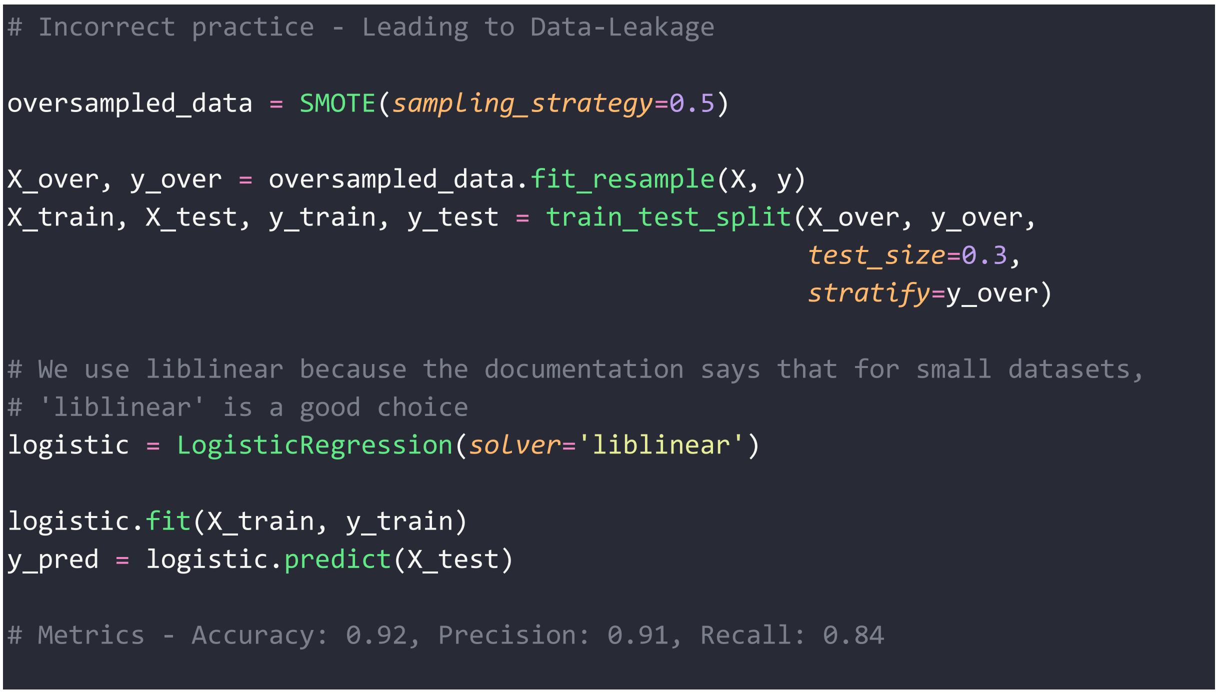
- SMOTE + Tomek Links
- SMOTE + Edited NearestNeighbors

```
from imblearn.combine import SMOTEENN

combined_sampling = SMOTEENN()

# Create model object
logistic = LogisticRegression(solver='liblinear')

# Initilaize pipeline with the required steps
pipeline = Pipeline(steps=[('combined', combined_sampling),
                           ('model', logistic)])
scoring = ['accuracy', 'precision', 'recall']

cv = RepeatedStratifiedKFold(n_splits=10, n_repeats=3, random_state=1)

# evaluating the model
scores = cross_validate(pipeline, X, y, scoring=scoring, cv=cv,
                        n_jobs=-1,  return_train_score=True)
print('Accuracy: {:.2f}, Precision: {:.2f}, Recall: {:.2f}'.format(
                                    np.mean(scores['test_accuracy']),
                                    np.mean(scores['test_precision']),
                                    np.mean(scores['test_recall'])))

# Metrics - Accuracy: 0.90, Precision: 0.51, Recall: 0.88
```

Here we can observe that `recall` is `0.88` and we have used Cross-Validation so the model is tested on entire data. So this result is always better than the result that we obtain from `train_test_split` as it is more trustful.

Incorrect practice

Sampling techniques or transformations are meant for train data and not for test or validation data.



```
# Incorrect practice - Leading to Data-Leakage

oversampled_data = SMOTE(sampling_strategy=0.5)

X_over, y_over = oversampled_data.fit_resample(X, y)
X_train, X_test, y_train, y_test = train_test_split(X_over, y_over,
                                                    test_size=0.3,
                                                    stratify=y_over)

# We use liblinear because the documentation says that for small datasets,
# 'liblinear' is a good choice
logistic = LogisticRegression(solver='liblinear')

logistic.fit(X_train, y_train)
y_pred = logistic.predict(X_test)

# Metrics - Accuracy: 0.92, Precision: 0.91, Recall: 0.84
```

Here we have over-sampled the data and then we did `train_test_split`, this leads to **'Data-leakage'**. We should not apply any sampling technique to test data, sampling techniques are meant only for train data. This is where `Pipelines` are useful, they are quite smart and they take care of such things. They would only apply those transformations to train data which are meant for train data, the same happens with test data as well, so we don't fall trap to **'Data-leakage'**.

Words of Wisdom

The title of the chapter might sound spiritual but it could cover some of the suggestions which we would like to give to the Data Science enthusiast.

Just learning isn't going to land you a job in Data Science, you need to gain experience while you work on projects so that it helps you in the interviews as well. When you start learning through a course the scenarios you face are the 'Good Good' scenarios where everything works smoothly. The data would be clean and you would be spoon-fed but this isn't how the things would be in the industry. When you go into industry, even before touching the dataset, you should gain some domain knowledge as it will help you in coming up with assumptions.

Investing 2-3 months on learning Python, Matplotlib, Pandas, NumPy, etc is not going to help as you won't be using all the things which you will learn. Rather you should start working on projects where your goal should be to first understand the problem statement and then deal with data. To start with, work on existing projects, understand the problem statement, analyze their solution by keeping yourself under the shoes of the person who has developed the solution. Once you have explored around 5 projects then you can pick up any problem statement of your choice and you would be in a position to work effectively on it.

This field is more about using our analytical skills and knowing what to apply and where to use, it isn't much about how to get it done because when it comes to coding we have the entire web to look for. Being a good programmer, or being an expert in Python isn't something that is needed, it's good to be an expert but that isn't the only skill needed. You should be able to apply your logic and should be clear with where to apply and where to apply, how to implement it when it comes to coding becomes easy by looking for the code on the web. And this is how all the developers work, they are good until and unless they have access to the internet and Stackoverflow.

In our opinion, you should not be wasting 2-3 months on learning Python rather you should learn Python while you work on projects. Making fancy visualizations won't help you unless and until you have some questions which you want to be answered out of those visualizations. Have some questions in mind and then plot a graph.

Let’s visualize `y=mx+c` in layman words:

Suppose you are learning something new or consider you are reading this book, there is something which you are looking for out of this book or else why would you waste your time.

y = What you expect out of this book i.e. looking at EDA from a completely new horizon and to learn something new.

x = The knowledge which you have prior reading this book (it is independent of the fact that you read this book or not, we also call it an independent feature)

m = Importance of the existing knowledge you have (it should have lesser value as you don’t need any pre-requisite to understand the things written in this book)

c = number of times you read this book and if stuck do you get in touch with the authors?

The day you start creating such analogies, it becomes very easy to understand the concepts.

If you don’t try to break down complex things into parts that are easy to understand it would become very difficult to understand complex things. So try to create some analogies and it would make you remember better. Or else when you would come across some text full of jargons, you would be like -

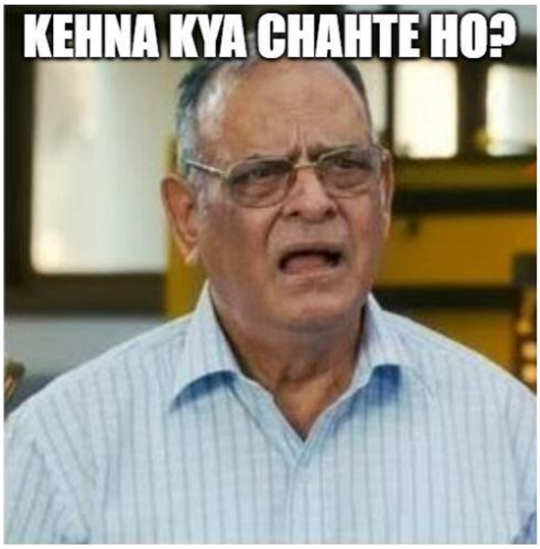


Figure 20: Kehna kya chahte ho? (humour)[1]

[1] Image Source - https://imgflip.com/

But once you make analogy you would feel –

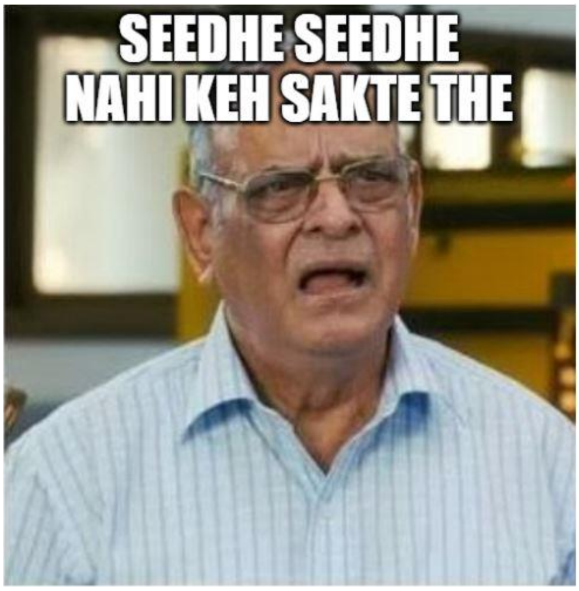


Figure 21: Seedhe Seedhe nahi keh sakte the (humour)[1]

The below analogy could help you to remember a real-life use-case of Hypothesis testing.

Suppose that your company thinks of changing the look and feel of their website so that they could attract more customers. You have to decide if the new look for your website is going to help your company to engage more customers than the current (old) website.

You cannot ask every customer of yours to visit the website and share the feedback but you can ask a few customers. In this case, you are dealing with a sample and you want to take a decision keeping the population in mind. This is a scenario where we could use Inferential statistics to come to a conclusion for the population while dealing with the sample.

In this scenario, hypothesis testing could help –

Null hypothesis – There is no difference in customer engagement after deploying the new website.

Alternate hypothesis – There is a difference in customer engagement after deploying the new website.

[1] Image Source - https://imgflip.com/

But this won’t give us a clear idea about which website is better. We will have to run some additional tests to come to a final conclusion.

You cannot just blindly replace an old website by a new one, this could have some serious consequences, and also you cannot interact with the entire population hence we go ahead with Hypothesis testing.

Every Data Scientist will first **understand the business problem** and then **apply statistics and probability knowledge** to find out the different insights to work with.

Having known that, there are many visualization tools out there in the market, you have to think, whether the job is going to be easy or difficult?

For sure the job is going to be more difficult and you have to be very smart to know how to make use of such tools.

Automation tools are made to reduce your time and effort but you should be in a position to think and take a call. If you don’t know why you are plotting a particular graph and what you are looking from particular visualization, then tools can’t help you out in any way.

What is feature engineering, when & how to apply?

Feature engineering is something that you can master with tons of practice. But how?

Whenever you are dealing with any problem statement, the first way to approach is to search for some existing research paper for the same, read blogs related to it and see if there is an existing solution available. By doing this first step, you can work on feature engineering with strong fundamentals.

There are many different techniques as well to deal with features. Let's say if you are having 200 features in your data set, then it's obvious that you don't have to use all features and here you have to work on selecting important features that would help you to build a scalable solution.

One of the major problems that you may be going through is – not knowing where to apply the concepts that you have learned. You believe or not, 90% of Data Science Enthusiast fail to apply statistical concepts while dealing with a problem statement - it is one of the stuff everyone out there is lagging and one of the major reasons as well for not getting job/internship.

Are you a Newbie and want to start with this domain?

Here are the steps to get in -

1. Start reading Blogs on a daily basis and try to understand the domain from the ground, try to find out some alternative solutions for the existing one. Maybe you won't get the stuff initially but when you work continuously, you will start thinking and understanding the stuff differently.
2. To be a Data Scientist, **Analytical Approach** is a must and you will get it after reading blogs and research papers related to the problem statement.
3. Maybe you are confused about which "Programming Language" you should master to start with. But **"The best programmer with less analytics thinking in terms of Statistics and Probability is not a good Data Scientist"**. So, the very first phase is to think like a "Data Scientist" and then take one step ahead with "Tech" to implement it. So, focus on Statistics and Probability with a **practical** as well as a **theoretical approach.**
4. Do not follow multiple stuff at the same time otherwise, you will get confused. Start with one, finish it, and research on the web for the problems and concepts you didn't get.

If you are a newbie then this is the right approach, follow it for a couple of months and after that, you would be able to see a difference in yourself. Most people out there focus on 'accuracy', it's important but 'model's simplicity and interpretability' is also important.

Before building any model, follow the below 3 steps to get to an optimal solution –

1. Think from the business perspective and make some hypotheses.
   Note: The hypothesis is just an assumption you make to deal with the business problem.
2. Think from the customer perspective and make some hypotheses.
3. Think from your perspective and make some hypotheses.

The hypothesis you come up with might be wrong but it would help you get a better understanding about the data.

Bot detection

Problem Statement: Build a model that can detect the Non-Human Traffic present on a website.

Wrong practice - As the problem statement seems quite easy so most of you will directly start working on dataset without understanding the business case in detail. That is where you would go wrong. First, we will spend some time making ourselves familiar with the problem statement.

Why do we need to detect Bots?

No matter how good your website is, you're always guaranteed to receive traffic from bots at some point in time. These bots can do different things on your website ranging from indexing web pages to scraping your content. With so many different bots out there, how can you detect bot traffic on your website? And should you be concerned about it?

Here are 5 reasons to Why you need Bot Detection?

- **Bots can steal your content -** You know that the content you worked upon is hard to develop. You have carefully crafted all the blog posts and pages, all the effort could be lost in a second if you let bots access your site. Bots can scrape your website for data, information, and even pricing in just a matter of time. This data can be used on other sites, redistributed, or even be sold to someone else.

- **Bots can slow down your site -** Bots bog down your site and overwhelm it with inauthentic, fraudulent traffic. This results in slower page load times for your actual paying customers, which could affect their level of satisfaction or even deter them from buying or visiting altogether.

- **Bots can threaten your website**. Malicious bots can hack your website, insert inappropriate links and content, or even crash your site altogether. This can hurt your traffic, your customers, and your sales.

- **A bot can take up extra time and money**. Many bots spend their time posting spam comments to websites and blogs. While this may not seem like a huge issue, it can be quite frustrating. You'll have to spend hours each month sorting through these comments to separate the human commenters from the fraudulent ones, which takes you and your resources away from actually running your business. If you don’t remove these spam comments, they end up annoying your readers and possibly leading them away from your site.

- **Bots can mess up your analytics**. Analytics is highly important to a website owner. They tell you how your site is performing, where traffic is coming from, and what you might want to tweak throughout the site. Unfortunately, if you have a significant amount of bots accessing your site, this can throw your analytics into upheaval. You won’t have a clear picture of your site’s performance or your next steps for improvement, and you won’t be able to tell what’s real and what’s fake.

So, we have done a bit of research about the problem statement and we have understood why is it important to detect a bot. It is advised that you research at your end as well, try to read about how can we detect bots, techniques used by different people out there. This would help you to develop an understanding of how should you proceed and will give you a starting point to think further.

Now we are going to look at how this problem can be solved by a wrong approach and then by the right approach using Feature Engineering and making assumptions.

Note - Main motive is to tell you the right approach to solve any project, we will focus on making assumptions and will not be discussing the code, it could be downloaded from the Github[1] link. So feel free to go through the code and you can get in touch with the authors on Linkedin for your doubts

Let’s have look at the Data Set:

You can download the .ipynb file from the Github[1] link and by looking at the code you would get an idea about the number of records, number of features, number of null records, and many more things.

Below are the features so that you can follow along.

```
['Unnamed: 0', 'ctry_name', 'intgrtd_mngmt_name',
'intgrtd_operating_team_name', 'city', 'st', 'sec_lvl_domn',
'device_type', 'operating_sys', 'ip_addr', 'user_agent', 'VISIT',
'ENGD_VISIT', 'VIEWS', 'page_url', 'wk', 'mth', 'yr', 'page_vw_ts']
```

[1] Github link - https://github.com/Dataebook/BotdetectionSnippet

Wrong Approach

At first, we will see how this problem statement can be dealt using the wrong approach as this is what most of us would do -

1. We might delete some columns which we think are not useful, this is the approach which most of them follow, if we have null values more than 40%, we just delete that column.

```
# dropping uselss columns & too many nulls
data.drop('Unnamed: 0',axis=1,inplace=True)
data.drop('device_type',axis=1,inplace=True)
```

2. We may observe that we have date column so we will convert it to appropriate format.

```
data.page_vw_ts = pd.to_datetime(data.page_vw_ts)
data['date'] = pd.to_datetime(data.page_vw_ts)
```

3. Then we may check for the different values of year and date.

   Now, we will again check for Nulls and we won't find any as we have dropped all the Nulls without understanding whether that particular column makes sense to our business problem or not.

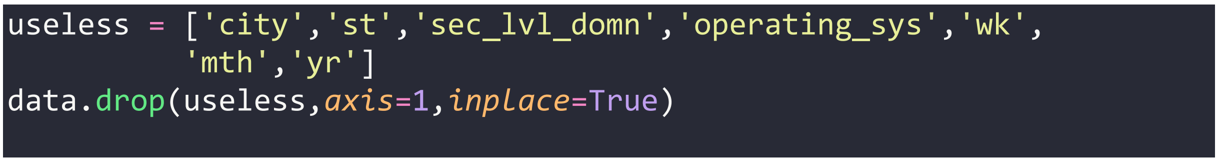
   We have a big dataset with millions of records and we can't work with the entire dataset at once, so we may check which country contributes to maximum traffic with the help of the below code.

```
plt.figure(figsize=(15,8))
sns.countplot(y=data['intgrtd_mngmt_name'])
plt.title('intgrtd_mngmt_name',size=20)
```

   After seeing the output we come to know that most of the IPs are from India, USA, and Japan.

4. Visualization we may look for -
   - Which is the most used OS?
   - Which Website is Visited Most?

5. Without any understanding about the problem statement, we may drop some columns as we might think they are not useful and we do this without any research



```
useless = ['city','st','sec_lvl_domn','operating_sys','wk',
          'mth','yr']
data.drop(useless,axis=1,inplace=True)
```

6. In the end, we might decide to work only with some of the top countries (traffic-wise). Post this we would proceed to build a model and our model would be able to detect when it's a bot but would fail to detect when it isn't a bot, which is important as well.

   We didn't observe whether our model is learning or memorizing. The saddest part is that we finished this project of millions of data points in just some hours without any research and are feeling overwhelmed by the fact that we have applied Machine Learning Algorithms.

Note - You can find the notebook with the wrong solution on Github[1].

[1] Github link - https://github.com/Dataebook/BotdetectionSnippet

Right Approach

- We will try to understand every feature and will try to understand their importance by doing some research on the internet.
- Also, check if there are some already available ways to solve this use-case as it could give us some idea and will enhance our thinking.
- We will try to understand the behavior of the bots and will try to observe it quite closely as it would help us in feature engineering and also to make assumptions.

Features Overview

`ctry_name` - Represents the country name.

`intgrtd mgmt name` - Represents management name it belongs to.

`intgrtd operating team_name` - Represents operating team name.

And somehow `ctryname`, `intgrtd mngmt name` and `intgrtd operating team` name don't make any sense concerning our problem statement, give it a thought?

`city` - Represents a city name.

`st` - Represents State name.

`sec lvl domn` - Represents Domain name.

`device_type` - Type of device.

`operating_sys` - Type of operating system.

`ip_addr` - IP of a particular device and this column plays a major role.

`user_agent` - Gives information about the browser that was used to hit the URL and version type of user agent.

`VISIT` - Tells the number of times a particular ip_address visited an URL.

`ENGD_VISIT` - Describes the engagement of a particular IP with the URL.

`VIEWS` - Calculate the number of views from particular ip_address on URL.

`Page url` - This feature is self-explanatory.

`wk` – Week info.

`mth` - Month info.

yr - Year info.

Page vw ts – Tells about the time when the page was visited.

Questions we might start asking -

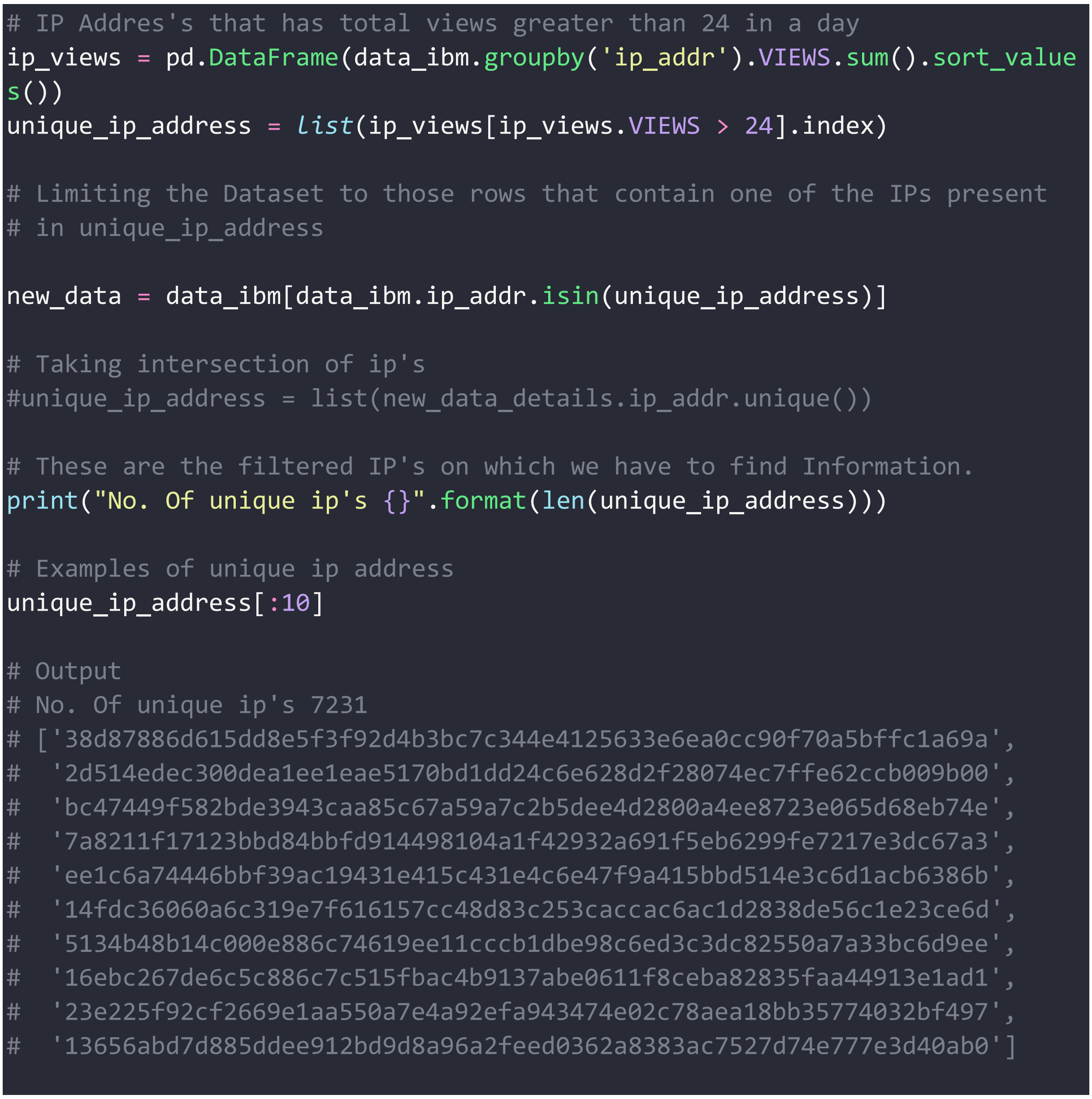
1. How IP address plays a major role to detect whether it is a bot or not?
2. Difference between `visit`, `engd_visit`, and `views`.
3. Importance of `visit, engd_visit,` and `views` while detecting a bot (understand the bot's behavior by research).
4. Does `user_agent` make sense while detecting bot, if yes, think how?
5. Difference between `wk, mth, yr,` and `page vw ts`, and are all the four features important?
6. If a user visits the URL via different `device_type` via different `operating_sys` will it imply somehow the behavior of a bot or not a bot?
7. Does `sec lvl domain` make any sense for our use-case?
8. Are `ctry_name, intgrtd mgmt name`, and `intgrtd operating team_name` important features or not?
9. As we know we have 10 million + records and for sure we can't deal with all IP's and at the same time we can have a lot of duplicates, so how will we deal with it?

   Note: You will get the required .ipynb file on Github[1].

   We can't work on all the IPs that visit our website in a day because many of them visit just once. So we try to filter those IPs that have high no. of views or visit too many times. In rules Of detecting bot, it is mentioned that bots show a similar pattern in visiting any website.

   We can assume that a normal human won't visit or view a particular website more than 24 times a day, just an assumption as the day is of 24 hours.

[1] Github link - https://github.com/Dataebook/BotdetectionSnippet



```
# IP Addres's that has total views greater than 24 in a day
ip_views = pd.DataFrame(data_ibm.groupby('ip_addr').VIEWS.sum().sort_value
s())
unique_ip_address = list(ip_views[ip_views.VIEWS > 24].index)

# Limiting the Dataset to those rows that contain one of the IPs present
# in unique_ip_address

new_data = data_ibm[data_ibm.ip_addr.isin(unique_ip_address)]

# Taking intersection of ip's
#unique_ip_address = list(new_data_details.ip_addr.unique())

# These are the filtered IP's on which we have to find Information.
print("No. Of unique ip's {}".format(len(unique_ip_address)))

# Examples of unique ip address
unique_ip_address[:10]

# Output
# No. Of unique ip's 7231
# ['38d87886d615dd8e5f3f92d4b3bc7c344e4125633e6ea0cc90f70a5bffc1a69a',
#  '2d514edec300dea1ee1eae5170bd1dd24c6e628d2f28074ec7ffe62ccb009b00',
#  'bc47449f582bde3943caa85c67a59a7c2b5dee4d2800a4ee8723e065d68eb74e',
#  '7a8211f17123bbd84bbfd914498104a1f42932a691f5eb6299fe7217e3dc67a3',
#  'ee1c6a74446bbf39ac19431e415c431e4c6e47f9a415bbd514e3c6d1acb6386b',
#  '14fdc36060a6c319e7f616157cc48d83c253caccac6ac1d2838de56c1e23ce6d',
#  '5134b48b14c000e886c74619ee11cccb1dbe98c6ed3c3dc82550a7a33bc6d9ee',
#  '16ebc267de6c5c886c7c515fbac4b9137abe0611f8ceba82835faa44913e1ad1',
#  '23e225f92cf2669e1aa550a7e4a92efa943474e02c78aea18bb35774032bf497',
#  '13656abd7d885ddee912bd9d8a96a2feed0362a8383ac7527d74e777e3d40ab0']
```

We will proceed with three Major steps.

Step 1 -

We have come up with a new feature known as **`Bounce_rate`** which implies how many times a particular IP has hit the URL on an hourly basis and it's the same as the number of vibrations per second known as frequency.

Also, we have created new features for the number of hours starting from `0_hour` to `23_hour` which would help us to visualize in which hour of the day a particular IP address visits the URL and will also help in monitoring the behavior of IP, for example, if one IP visits the URL more than 15 times in an hour then we can assume it can be a bot.

Just give a thought to what you just read.

We can then look for the information about the unique IP's, this will give info about visits per hour, the corresponding Bounce rate of each IP and also information related to its origin and using this all information we can construct a dataframe on which we can train a model.

We have introduced some more features like `hour_avg` and `daily_avg` which helps us to identify how often that particular IP visits the URL in a particular hour and based on that we will calculate the `bounce_rate`.

Download the .ipynb file from Github for better understanding.

As this book is something where you would have to feel the pain so give it a thought, do some research about what we are discussing, and get back to us for any queries.

"No Pain, No Gain"

We have created a new dataset only for unique IP's addresses and we will build a model using this dataset.

So far, we have two datasets one is original and another is with unique IP's addresses.

Step 2 -

Here we will make a dataset called Global Dataset, basically, Global Dataset will contain all the history of an `ip_addr` that has visited us before and if it visits us again then this will append its important values to the `global_dataset`. This Dataset will be the most important part of the program because it will help in labeling the classes as a bot, it contains the information that will be used by our next part when we label the classes.

Importance of 'Global Dataset' -

For example, if you productionize the model, and imagine an IP comes in then:

1. It will be compared with the existing dataset, if that particular IP address has visited us earlier at some point in time, then based on the past behavior of the IP address the model will decide whether it is a bot or not.

2. What if the particular IP address is not present in past data, then it won't be sent directly to the model. At first, it will be sent to the global dataset and where its daily, hourly, and weekend activity is calculated. Also, we have created some attributes like `hourly_avg` and `daily_avg` to calculate the exact bounce rate based on which model will predict whether it's a bot or not.

   Just think about this

   Let's say you have created a new attribute **`bounce_rate`** and with the help of `hourly_avg` and `daily_avg` model we will calculate the exact **`bounce_rate`**. But when you productionize the model the unseen data or the data coming in won't have the new features that you have created with the help of research and understanding. So before feeding data to the model you have to work with custom code or create your pipeline which can convert the original attribute at production to the attributes that you created using feature engineering (feature engineering is something which is not defined and your research about the business problem with common sense helps to get it done).

   So, how will you tackle such things?

Step 3 -

In this part, we will label the IPs in the "`new_ip_data`" dataset with the help of ''Global Dataset" and the Rules.

With the help of research and understanding the bot behavior we have come up with some rules which will help our model to achieve a scalable solution.

We will write a function that will return a dataframe that has all the historical data of a single IP address (Don't worry, download the .ipynb file from Github and play around with the code to make your self comfortable).

In the end, after a lot of research (which takes time with patience), understanding, and commonsense, we have come up with some rules and better ways to understand how your model should work at production.

So, give it a thought and analyze how these rules can make our model learn incrementally.

We don't believe in spoon-feeding, perform your **EDA** for the same and build your assumptions as there are multiple solutions for the same, do get in touch with us for more discussion on the same at any point of time.

Rule 1 - Labeling based on hour

Rule 2 - Labeling on the basis of `daily_avg`

Rule 3 - Labeling on the basis of `weekday_avg`

Rule 4 - Labeling on the basis of bounce rate

Rule 5 - Labeling on the basis of Operating system

If we don't have past information of an `ip_address` then we are left with only three rules.

- **`Hourly_average`**
- **`Bounce_rate`**
- **`Operating_system`**

So, we have come up with a lot of assumptions, do download the .ipynb file from the Github[1]. Try, fail, build, and discuss with us at any point in time.

Remember this isn't the only possible solution, there could be many other solutions available and they could be found by doing some research and taking efforts to solve this use-case.

[1] Github link - https://github.com/Dataebook/BotdetectionSnippet

FROM THE AUTHORS DESK

Vivek Chaudhary

Creator at **Dataebook || Data Scientist || Community Builder**

Linkedin[1]

Around August '17, a question popped up in my mind about how companies like Samsung fix the price of their new phones & are confident enough to get a good margin of profit from the market.

That hit my mind & then I started researching about the same and that was my first step into the Data Science domain.

From that day, on a daily basis, I used to read different case studies about how Data Science is helping industries to do better, which in turn helped me to understand this domain closely to get expertise with.

You have to be a research enthusiast before getting into this domain & be clear about why you want to get into this domain, as Data Science is not Everyone's cup of tea.

Some of the advice I would love to deliver to the readers,

1. Don't compare yourself with other people in the same domain, if you do then you doubt yourself & lead yourself to nowhere except negative thoughts.
2. Sometimes we may think that Kaggle is something that is out of the box & you can't compete with Kaggle Grandmasters but did you ever think, once those masters were also at your position and had the same thinking. But they didn't give up, patience & enthusiasm about research is something that helped them to achieve it.

   "No-one is born as an expert but one can die as an expert. You just have to work with patience because good things take time".

[1] Linkedin Profile - https://www.linkedin.com/in/-vivek-chaudhary

3. Don’t focus on building a Machine Learning model with 95% accuracy, instead focus on the process to build a Machine Learning model which starts from understanding the business case, making assumptions, EDA, applied statistics, data cleaning technique, feature encoding technique, preprocessing before building a model.
4. Don’t waste your time learning python for data science, instead choose any industry, research about how Data Science is helping out that industry/organization & look at the existing solution and then figure out if there something new you can come up with.

   Remember, if you research, only then you can come up with your thoughts otherwise you will be busy with building Machine Learning models with 95% accuracy which is of no use for your client.

5. Pick any existing project with a solution, understand it, and try to analyze the code.
   - Don’t worry if you don’t understand any line of code, just research about the same & learn about it while you apply.
   - For sure it requires your patience & hard work because some of the time you may have to invest 4 to 5 hours to understand a couple of lines of codes but believe me if you follow this process for at least 2 to 4 projects then you would definitely get good at it.

6. If you want to get into this domain, first you have to believe in yourself because if you don’t, then no one will. You can't master this domain as new things come up every now and then, but yes, if you follow such an approach then for sure, you will be confident enough to work in a project because you have learnt it in a hard way.
7. Don’t make yourself limited to a certain technique, for example, most of the individual will apply one-hot encoding or label encoding while dealing with categorical features without knowing the importance. Instead if you search, you would find a lot of different techniques & if we understand the concept, then who knows whether we may come up with some new techniques to deal with.

Don’t be in a rush, because this domain needs patience and a lot of smart as well as hard work.

Thanks for reading & you can get in touch with me at any point of time for more discussion on the same.

Anirudh Dayma

Machine Learning Engineer | Technical Writer

Linkedin[1] | Medium [2]

I love to explore this field of Data Science, I also write technical articles for Analytics Vidhya and Towards AI. My aim is to learn new things and explain them in the simplest possible way. As Albert Einstein has rightly said that "If you can't explain it to a 6-year-old, you don't understand it yourself". I truly believe in this statement of his and hence try my best to explain stuff in a simpler way.

I believe that Data Science is a technique rather than a domain, it is a strategy that could be applied to any domain. The way we have the best coding practices/techniques which could be used in any programming language the same goes with Data Science as well, we are free to use it in any domain of our choice. Like all the readers of this book who are here to learn something from this book similarly, I am learning each day from everyone around me.

I was fascinated by this field by looking at the wonders it could do and so thought of exploring this field. At first, even I wasted my time learning R for Data Science, Python for Data Science, and in the end, I was unable to remember anything I learned. So the reason I was unable to remember the Python or R concepts/code snippets was because I was just gathering knowledge and not applying what I learned. So I started applying the things I learnt by working on small projects and applying Machine learning algorithms by understanding the maths behind them.

[1] Linkedin Profile - https://www.linkedin.com/in/anirudh-dayma-457861144/

[2] Medium Profile - https://medium.com/@anirudh.daymaa

Post this I started exploring Statistics when I realized the importance of making assumptions and it led me to Hypothesis Testing. Statistics is the topic which many of the Data Science Enthusiasts skip because they don’t find it interesting also there are not many resources that explain Statistics in a simple manner. We have tried to explain statistics by sighting some real-life examples so that after reading this book even if you don’t remember the definition, you would remember the example which would help you to remember the concept.

So my suggestion is please do focus on Maths as Data Science is all about Linear Algebra, Calculus, Statistics. Many people tend to run away from these things but take my words there is a lot of Automation happening around us, so to stay in the market we should be able to understand the intuition behind the things we are learning. Making models isn’t difficult these days as there are many tools readily available to do that. Focus on the essence behind the jargon/concepts, ask as many WHY’s as possible, these WHY’s would open new doors for you and make you understand things in a better way.

Manvendra Singh

ML Developer Codegnan IT Solutions

Linkedin[1]

Hi, myself Manvendra Singh working as a Machine Learning Developer at Codegnan IT Solutions. Data Science is a journey.

Data Science is a process to solve business problems through data. It should not be confused with any technology. The technology that we use is just the tools that we have at our disposal to solve these problems. During the initial years when I started my Data Science journey, I had this false assumption about Data Science that it is just building fancy models. I have learned the hard way that Data Science is not about building all the fancy models that are there. It is about being able to understand the problem statement and using your common sense, technology and skills to solve that problem.

How to master Data Science? Well, I am still looking for that person who says that he has mastered Data Science. You cannot master Data Science, no one can, it is a process where you apply your knowledge to solve the problem at hand, and one problem may have different solutions, there is no such streamlined process that will make you the master of data science. So try to master problem-solving and critical thinking skills. If you are a good problem solver you can be a good Data Scientist too. And to master problem-solving skills you have to get your hands dirty.

In the first phase, I used to learn algorithms and apply them to some dataset. And this kept on going for quite some time. When I talked with some of the

[1] Linkedin Profile - https://www.linkedin.com/in/me-manvendra/

industry experts then I realized, no one cares about your fancy algorithms. They want to see how you solved a given problem statement and what value it adds to society and the company. Then started solving the problem statements, not building a fancy classifier. And when you work on these problem statements and the projects and get your hands dirty that way you learn to solve problems, you start with analytical thinking and reasoning.

So my advice to anyone who is looking to start their career in this field would be to work on projects. The learning you will have from working on projects cannot be compared with other programs. You will become a better problem solver and all the technology will be just a tool for you to solve these problems.

John Gabriel T J

Data Analyst || Data Science Enthusiast

Linkedin[1]

I am John Gabriel T J and I'm an ECE graduate. I started my career as a Software developer in a startup company and worked there for 6 months. Then I worked as a Technical Support Executive in a BPO company for almost 1 year. After that, I moved to another Automobile company, working as a Data Analyst.

If you could see my transition, it is nowhere connected to each other, from my graduation. But one thing, what I can tell you is that I was struggling to find what suits me best to keep me motivated while working. To keep the long story short, finally, I found Data Science to be the right fit for me not because it is a booming industry but by the work it has.

In a quick note, a Data Scientist is someone who can predict the future based on past patterns. Who wouldn't be glad and interested to work to make changes to the world we live in, with the help of data?

I started learning Data Science with R programming language as people suggested. I was not at all able to grasp anything at the beginning time. I had no idea what I was learning also. I was confused with all the Statistical terms and concepts. But eventually, as I kept learning, things got changed. I motivate myself whenever I go down by watching and realizing the new revolution that AI creates and impacts the society that we are living every day.

[1] Linkedin Profile - https://www.linkedin.com/in/johngabrields/

Because it’s so easy to get frustrated while learning new things, not only Data Science. Without proper motivation, it would be very difficult for anyone to continue learning new stuff. However, in my case, I started understanding concepts and was able to learn better as I kept engaging myself.

Along the way, I’ve attended a few sessions of Vivek. After attending his few lectures, I was able to learn things from a different perspective. All the stuff I learnt until that time, I was only learning and nowhere applied. After trying out his suggested way of learning, I can now say that it is more practical and more interesting as well. All he said was, learn Data Science in a reverse engineering method.

I started learning Machine Learning through small datasets taken by Kaggle projects. And one of the projects that I’ve worked, is published in this book. I’m still working on many other areas as well. Due to certain limitations, I am not able to publish it all.

My motivation:

“Learn from mistakes and with the lesson what you learnt, apply it in daily life to overcome it. Don’t wish for it. Do it”.

Though my career has started from being a Developer, I’ve now found what work suits me best and working on it to achieve it very seriously.

Leave a review

Please share your thoughts on this book by leaving a review on the site that you bought it from. This would help other potential readers to make purchasing decisions.

Incase of any queries or corrections reach out to the authors on Linkedin.