INTRODUCTION TO DATA SCIENCE

LECTURE NOTES

UNIT - 1

Introduction to data science

Data science:

Data science is the domain of study that deals with vast volumes of data using modern tools and techniques to find unseen patterns, derive meaningful information, and make business decisions. Data science uses complex machine learning algorithms to build predictive models.

The data used for analysis can come from many different sources and presented in various formats.

Data science is about extraction, preparation, analysis, visualization, and maintenance of information. It is a cross disciplinary field which uses scientific methods and processes to draw insights from data.

The Data Science Lifecycle

Data science’s lifecycle consists of five distinct stages, each with its own tasks:

Capture: Data Acquisition, Data Entry, Signal Reception, Data Extraction. This stage involves gathering raw structured and unstructured data.

Maintain: Data Warehousing, Data Cleansing, Data Staging, Data Processing, Data Architecture. This stage covers taking the raw data and putting it in a form that can be used.

Process: Data Mining, Clustering/Classification, Data Modeling, Data Summarization. Data scientists take the prepared data and examine its patterns, ranges, and biases to determine how useful it will be in predictive analysis.

Analyze: Exploratory/Confirmatory, Predictive Analysis, Regression, Text Mining, Qualitative Analysis. Here is the real meat of the lifecycle. This stage involves performing the various analyses on the data.

Communicate: Data Reporting, Data Visualization, Business Intelligence, Decision Making. In this final step, analysts prepare the analyses in easily readable forms such as charts, graphs, and reports.

Evolution of Data Science: Growth & Innovation

Data science was born from the idea of merging applied statistics with computer science. The resulting field of study would use the extraordinary power of modern computing. Scientists realized they could not only collect data and solve statistical problems but also use that data to solve real-world problems and make reliable fact-driven predictions.

1962: American mathematician John W. Tukey first articulated the data science dream. In his now-famous article "The Future of Data Analysis," he foresaw the inevitable emergence of a new field nearly two decades before the first personal computers. While Tukey was ahead of his time, he was not alone in his early appreciation of what would come to be known as "data science."

1977: The theories and predictions of "pre" data scientists like Tukey and Naur became more concrete with the establishment of The International Association for Statistical Computing (IASC), whose mission was "to link traditional statistical methodology, modern computer technology, and the knowledge of domain experts in order to convert data into information and knowledge."

1980s and 1990s: Data science began taking more significant strides with the emergence of the first Knowledge Discovery in Databases (KDD) workshop and the founding of the International Federation of Classification Societies (IFCS).

1994: Business Week published a story on the new phenomenon of "Database Marketing." It described the process by which businesses were collecting and leveraging enormous amounts of data to learn more about their customers, competition, or advertising techniques.

1990s and early 2000s: We can clearly see that data science has emerged as a recognized and specialized field. Several data science academic journals began to circulate, and data science proponents like Jeff Wu and William S. Cleveland continued to help develop and expound upon the necessity and potential of data science.

2000s: Technology made enormous leaps by providing nearly universal access to internet connectivity, communication, and (of course) data collection.

2005: Big data enters the scene. With tech giants such as Google and Facebook uncovering large amounts of data, new technologies capable of processing them became necessary. Hadoop rose to the challenge, and later on Spark and Cassandra made their debuts.

2014: Due to the increasing importance of data, and organizations' interest in finding patterns and making better business decisions, demand for data scientists began to see dramatic growth in different parts of the world.

2015: Machine learning, deep learning, and Artificial Intelligence (AI) officially enter the realm of data science.

2018: New regulations in the field are perhaps one of the biggest aspects in the evolution in data science.

2020s: We are seeing additional breakthroughs in AI, machine learning, and an ever-more-increasing demand for qualified professionals in Big Data

Roles in Data Science

Data Analyst

Data Engineers

Database Administrator

Machine Learning Engineer

Data Scientist

Data Architect

Statistician

Business Analyst

Data and Analytics Manager

1. Data Analyst

Data analysts are responsible for a variety of tasks including visualisation, munging, and processing of massive amounts of data. They also have to perform queries on the databases from time to time. One of the most important skills of a **data analyst** is optimization.

Few Important Roles and Responsibilities of a Data Analyst include:

Extracting data from primary and secondary sources using automated tools

Developing and maintaining databases

Performing data analysis and making reports with recommendations

To become a data analyst: SQL, R, SAS, and Python are some of the sought-after technologies for data analysis.

2. Data Engineers

Data engineers build and test scalable Big Data ecosystems for the businesses so that the **data scientists** can run their algorithms on the data systems that are stable and highly optimized. **Data engineers** also update the existing systems with newer or upgraded versions of the current technologies to improve the efficiency of the databases.

Few Important Roles and Responsibilities of a Data Engineer include:

Design and maintain data management systems

Data collection/acquisition and management

Conducting primary and secondary research

To become data engineer: technologies that require hands-on experience include Hive, NoSQL, R, Ruby, Java, C++, and Matlab.

3. Database Administrator

The job profile of a database administrator is pretty much self-explanatory- they are responsible for the proper functioning of all the databases of an enterprise and grant or revoke its services to the employees of the company depending on their requirements.

Few Important Roles and Responsibilities of a Database Administrator include:

- Working on database software to store and manage data
- Working on database design and development
- Implementing security measures for database
- Preparing reports, documentation, and operating manuals

To become database administrator: database backup and recovery, data security, data modeling, and design, etc

4. Machine Learning Engineer

Machine learning engineers are in high demand today. However, the job profile comes with its challenges. Apart from having in-depth knowledge of some of the most powerful technologies such as SQL, REST APIs, etc. machine learning engineers are also expected to perform A/B testing, build data pipelines, and implement common machine learning algorithms such as classification, clustering, etc.

Few Important Roles and Responsibilities of a Machine Learning Engineer include:

- Designing and developing Machine Learning systems
- Researching Machine Learning Algorithms

- Testing Machine Learning systems
- Developing apps/products basis client requirements

To become machine learning engineer: technologies like Java, Python, JS, etc. Secondly, you should have a strong grasp of statistics and mathematics.

5. Data Scientist

Data scientists have to understand the challenges of business and offer the best solutions using data analysis and data processing. For instance, they are expected to perform predictive analysis and run a fine-toothed comb through an “unstructured/disorganized” data to offer actionable insights.

Few Important Roles and Responsibilities of a Data Scientist include:

- Identifying data collection sources for business needs
- Processing, cleansing, and integrating data
- Automation data collection and management process
- Using Data Science techniques/tools to improve processes

To become a data scientist, you have to be an expert in R, MatLab, SQL, Python, and other complementary technologies.

6. Data Architect

A data architect creates the blueprints for data management so that the databases can be easily integrated, centralized, and protected with the best security measures. They also ensure that the data engineers have the best tools and systems to work with.

Few Important Roles and Responsibilities of a Data Architect include:

- Developing and implementing overall data strategy in line with business/organization
- Identifying data collection sources in line with data strategy
- Collaborating with cross-functional teams and stakeholders for smooth functioning of database systems

- Planning and managing end-to-end data architecture

To become a data architect: requires expertise in data warehousing, data modelling, extraction transformation and loan (ETL), etc. You also must be well versed in Hive, Pig, and Spark, etc.

7. Statistician

A statistician, as the name suggests, has a sound understanding of statistical theories and data organization. Not only do they extract and offer valuable insights from the data clusters, but they also help create new methodologies for the engineers to apply.

Few Important Roles and Responsibilities of a Statistician include:

- Collecting, analyzing, and interpreting data
- Analyzing data, assessing results, and predicting trends/relationships using statistical methodologies/tools
- Designing data collection processes

To become a statistician: SQL, data mining, and the various machine learning technologies.

8. Business Analyst

The role of **business analysts** is slightly different than other data science jobs. While they do have a good understanding of how data-oriented technologies work and how to handle large volumes of data, they also separate the high-value data from the low-value data.

Few Important Roles and Responsibilities of a Business Analyst include:

- Understanding the business of the organization
- Conducting detailed business analysis – outlining problems, opportunities, and solutions
- Working on improving existing business processes

To become business analyst: understanding of business finances and **business intelligence**, and also the IT technologies like data modelling, data visualization tools, etc.

Stages in a data science project

Data Science workflows tend to happen in a wide range of domains and areas of expertise such as biology, geography, finance or business, among others. This means that Data Science projects can take on very different challenges and focuses resulting in very different methods and data sets being used. A Data Science project will have to go through five key stages: defining a problem, data processing, modelling, evaluation and deployment.

Defining a problem

- The first stage of any Data Science project is to identify and define a problem to be solved. Without a clearly defined problem to solve, it can be difficult to know how to tackle to the problem.
- For a Data Science project this can include what method to use, such as is classification, regression or clustering. Also, without a clearly defined problem, it can be hard to determine what your measure of success would be.
- Without a defined measure of success, you can never know when your project is complete or is good enough to be used in production.
- A challenge with this is being able to define a problem small enough that it can be solved/tackled individually.

Data Processing

- Once you have your problem, how you are going to measure success, and an idea of the methods you will be using, you can then go about performing the all important task of data processing. This is often the stage that will take the longest in any Data Science project and can regularly be the most important stage.
- There are a variety of tasks that need to occur at this stage depending on what problem you are going to tackle. The first is often finding ways to create or capture data that doesn’t exist yet.

- Once you have created this data, you then need to collect it somewhere and in a format that is useful for your model. This will depend on what method you will be using in the modelling phase but it will involve figuring out how you will feed the data into your model.
- The final part of this is to then perform any pre-processing steps to ensure that the data is clean enough for the modelling method to work. This may involve removing outliers, or choosing to keep them, manipulating null values, whether a null value is a measure or whether it should be imputed to the average, or standardising the measures.

Modelling

- The next part, and often the most fun and exciting part, is the modelling phase of the Data Science project. The format this will take will depend primarily on what the problem is and how you defined success in the first step, and secondarily on how you processed the data.
- Unfortunately, this is often the part that will take the least amount of time of any Data Science project, especially when there are many frameworks or libraries that exist, such as sklearn, statsmodels, tensorflow and that can be readily utilised.
- You should have selected the method that you will be using to model your data in the defining a problem stage, and this may include simple graphical exploration, regression, classification or clustering.

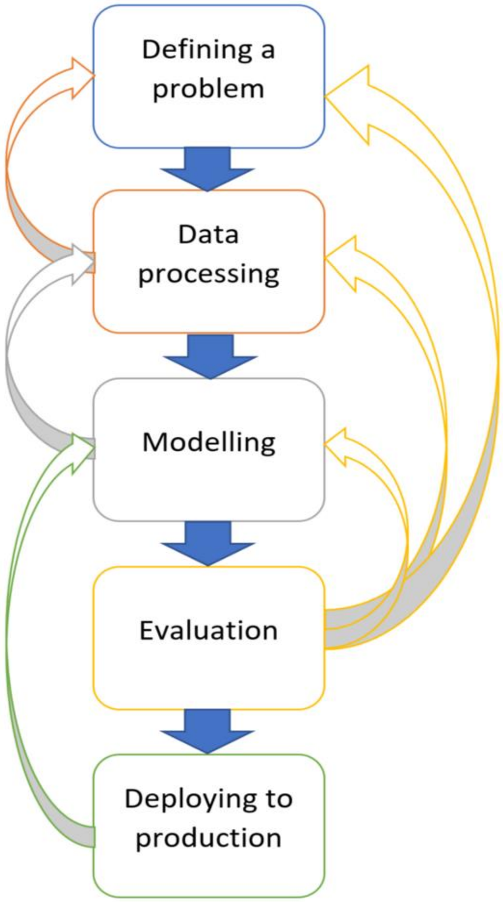
Evaluation

- Once you have then created and implemented your models, you then need to know how to evaluate it. Again, this goes back to the problem formulation stage where you will have defined your measure of success, but this is often one of the most important stages.
- Depending on how you processed your data and set-up your model, you may have a holdout dataset or testing data set that can be used to evaluate your model. On this dataset,

you are aiming to see how well your model performs in terms of both accuracy and reliability.

Deployment

Finally, once you have robustly evaluated your model and are satisfied with the results, then you can deploy it into production. This can mean a variety of things such as whether you use the insights from the model to make changes in your business, whether you use your model to check whether changes that have been made were successful, or whether the model is deployed somewhere to continually receive and evaluate live data.



Applications of data science in various fields

Major Applications of Data Science

1. In Search Engines
The most useful application of Data Science is Search Engines. As we know when we want to search for something on the internet, we mostly used Search engines like Google, Yahoo, Safari, Firefox, etc. So Data Science is used to get Searches faster.

2. In Transport
Data Science also entered into the Transport field like Driverless Cars. With the help of Driverless Cars, it is easy to reduce the number of Accidents.

For Example, In Driverless Cars the training data is fed into the algorithm and with the help of Data Science techniques, the Data is analyzed like what is the speed limit in Highway, Busy Streets, Narrow Roads, etc. And how to handle different situations while driving etc.

3. In Finance
Data Science plays a key role in Financial Industries. Financial Industries always have an issue of fraud and risk of losses. Thus, Financial Industries needs to automate risk of loss analysis in order to carry out strategic decisions for the company. Also, Financial Industries uses Data Science Analytics tools in order to predict the future.

For Example, In Stock Market, Data Science is the main part. In the Stock Market, Data Science is used to examine past behavior with past data and their goal is to examine the future outcome.

4. In E-Commerce
E-Commerce Websites like Amazon, Flipkart, etc. uses data Science to make a better user experience with personalized recommendations.

For Example, When we search for something on the E-commerce websites we get suggestions similar to choices according to our past data and also we get recommendations according to most buy the product, most rated, most searched, etc. This is all done with the help of Data Science.

5. In Health Care
In the Healthcare Industry data science act as a boon. Data Science is used for:

- Detecting Tumor.
- Drug discoveries.
- Medical Image Analysis.
- Virtual Medical Bots.
- Genetics and Genomics.
- Predictive Modeling for Diagnosis etc.

6. Image Recognition

Currently, Data Science is also used in Image Recognition. **For Example,** When we upload our image with our friend on Facebook, Facebook gives suggestions Tagging who is in the picture. This is done with the help of machine learning and Data Science. When an Image is Recognized, the data analysis is done on one's Facebook friends and after analysis, if the faces which are present in the picture matched with someone else profile then Facebook suggests us auto-tagging.

7. Targeting Recommendation

Targeting Recommendation is the most important application of Data Science. Whatever the user searches on the Internet, he/she will see numerous posts everywhere.

example: Suppose I want a mobile phone, so I just Google search it and after that, I changed my mind to buy offline. Data Science helps those companies who are paying for Advertisements for their mobile. So everywhere on the internet in the social media, in the websites, in the apps everywhere I will see the recommendation of that mobile phone which I searched for. So this will force me to buy online.

8. Airline Routing Planning

With the help of Data Science, Airline Sector is also growing like with the help of it, it becomes easy to predict flight delays. It also helps to decide whether to directly land into the destination or take a halt in between like a flight can have a direct route from Delhi to the U.S.A or it can halt in between after that reach at the destination.

9. Data Science in Gaming

In most of the games where a user will play with an opponent i.e. a Computer Opponent, data science concepts are used with machine learning where with the help of past data the Computer will improve its performance. There are many games like Chess, EA Sports, etc. will use Data Science concepts.

10. Medicine and Drug Development

The process of creating medicine is very difficult and time-consuming and has to be done with full disciplined because it is a matter of Someone's life. Without Data Science, it takes lots of time, resources, and finance or developing new Medicine or drug but with the help of Data Science, it becomes easy because the prediction of success rate can be easily determined based on biological data or factors. The algorithms based on data science will forecast how this will react to the human body without lab experiments.

11. In Delivery Logistics

Various Logistics companies like DHL, FedEx, etc. make use of Data Science. Data Science helps these companies to find the best route for the Shipment of their Products, the best time suited for delivery, the best mode of transport to reach the destination, etc.

12. Autocomplete

AutoComplete feature is an important part of Data Science where the user will get the facility to just type a few letters or words, and he will get the feature of auto-completing the line. In Google Mail, when we are writing formal mail to someone so at that time data science concept of Autocomplete feature is used where he/she is an efficient choice to auto-complete the whole line. Also in Search Engines in social media, in various apps, AutoComplete feature is widely used.

Data security issues

What is Data Security?

Data security is the process of protecting corporate data and preventing data loss through unauthorized access. This includes protecting your data from attacks that can encrypt or destroy data, such as ransomware, as well as attacks that can modify or corrupt your data. Data security also ensures data is available to anyone in the organization who has access to it.

Some industries require a high level of data security to comply with data protection regulations. For example, organizations that process payment card information must use and store payment card data securely, and healthcare organizations in the USA must secure private health information (PHI) in line with the HIPAA standard.

Data Security vs Data Privacy

Data privacy is the distinction between data in a computer system that can be shared with third parties (non-private data), and data that cannot be shared with third parties (private data). There are two main aspects to enforcing data privacy:

- **Access control**—ensuring that anyone who tries to access the data is authenticated to confirm their identity, and authorized to access only the data they are allowed to access.
- **Data protection**—ensuring that even if unauthorized parties manage to access the data, they cannot view it or cause damage to it. Data protection methods ensure encryption, which prevents anyone from viewing data if they do not have a private encryption key, and data loss prevention mechanisms which prevent users from transferring sensitive data outside the organization.

Data security has many overlaps with data privacy. The same mechanisms used to ensure data privacy are also part of an organization’s data security strategy.

The primary difference is that data privacy mainly focuses on keeping data confidential, while data security mainly focuses on protecting from malicious activity.

Data Security Risks

❖ Accidental Exposure

A large percentage of data breaches are not the result of a malicious attack but are caused by negligent or accidental exposure of sensitive data. It is common for an organization's employees to share, grant access to, lose, or mishandle valuable data, either by accident or because they are not aware of security policies.

❖ Phishing and Other Social Engineering Attacks

Social engineering attacks are a primary vector used by attackers to access sensitive data. They involve manipulating or tricking individuals into providing private information or access to privileged accounts.

Phishing is a common form of social engineering. It involves messages that appear to be from a trusted source, but in fact are sent by an attacker.

❖ Insider Threats

Insider threats are employees who inadvertently or intentionally threaten the security of an organization's data. There are three types of insider threats:

- **Non-malicious insider**—these are users that can cause harm accidentally, via negligence, or because they are unaware of security procedures.
- **Malicious insider**—these are users who actively attempt to steal data or cause harm to the organization for personal gain.
- **Compromised insider**—these are users who are not aware that their accounts or credentials were compromised by an external attacker. The attacker can then perform malicious activity, pretending to be a legitimate user.

❖ Ransomware

Ransomware is a major threat to data in companies of all sizes. Ransomware is malware that infects corporate devices and encrypts data, making it useless without the decryption key.

Attackers display a ransom message asking for payment to release the key, but in many cases, even paying the ransom is ineffective and the data is lost.

- **Data Loss in the Cloud**

Many organizations are moving data to the cloud to facilitate easier sharing and collaboration. However, when data moves to the cloud, it is more difficult to control and prevent data loss. Users access data from personal devices and over unsecured networks. It is all too easy to share a file with unauthorized parties, either accidentally or maliciously.

- **SQL Injection**

SQL injection (SQLi) is a common technique used by attackers to gain illicit access to databases, steal data, and perform unwanted operations. It works by adding malicious code to a seemingly innocent database query.

Common Data Security Solutions and Techniques:

Data Discovery and Classification

- Modern IT environments store data on servers, endpoints, and cloud systems. Visibility over data flows is an important first step in understanding what data is at risk of being stolen or misused.
- To properly protect your data, you need to know the type of data, where it is, and what it is used for. Data discovery and classification tools can help.
- Data detection is the basis for knowing what data you have. Data classification allows you to create scalable security solutions, by identifying which data is sensitive and needs to be secured.

Data Masking

- Data masking lets you create a synthetic version of your organizational data, which you can use for software testing, training, and other purposes that don’t require the real data.
- The goal is to protect data while providing a functional alternative when needed.

Data Encryption

- Data encryption is a method of converting data from a readable format (plaintext) to an unreadable encoded format (ciphertext). Only after decrypting the encrypted data using the decryption key, the data can be read or processed.

- In public-key cryptography techniques, there is no need to share the decryption key – the sender and recipient each have their own key, which are combined to perform the encryption operation. This is inherently more secure.

- Data encryption can prevent hackers from accessing sensitive information.

Password Hygiene

- One of the simplest best practices for data security is ensuring users have unique, strong passwords. Without central management and enforcement, many users will use easily guessable passwords or use the same password for many different services.

- Password spraying and other brute force attacks can easily compromise accounts with weak passwords.

Authentication and Authorization

Organizations must put in place strong authentication methods, such as OAuth for web-based systems. It is highly recommended to enforce multi-factor authentication when any user, whether internal or external, requests sensitive or personal data.

UNIT –II

DATA COLLECTION AND PREPROCESSING

DATA COLLECTION:

Data collection is the process of collecting, measuring and analyzing different types of information using a set of standard validated techniques. The main objective of data collection is to gather information-rich and reliable data, and analyze them to make critical business decisions. Once the data is collected, it goes through a rigorous process of data cleaning and data processing to make this data truly useful for businesses.

There are two main methods of data collection in research based on the information that is required, namely:

- Primary Data Collection
- Secondary Data Collection

Primary Data Collection Methods

Primary data refers to data collected from first-hand experience directly from the main source. It refers to data that has never been used in the past. The data gathered by primary data collection methods are generally regarded as the best kind of data in research.

- The methods of collecting primary data can be further divided into quantitative data collection methods (deals with factors that can be counted) and qualitative data collection methods (deals with factors that are not necessarily numerical in nature).

Here are some of the most common primary data collection methods:

1. Interviews

Interviews are a direct method of data collection. It is simply a process in which the interviewer asks questions and the interviewee responds to them. It provides a high degree of flexibility because questions can be adjusted and changed anytime according to the situation.

2. Observations

In this method, researchers observe a situation around them and record the findings. It can be used to evaluate the behaviour of different people in controlled (everyone knows they are being observed) and uncontrolled (no one knows they are being observed) situations.

3. Surveys and Questionnaires

Surveys and questionnaires provide a broad perspective from large groups of people. They can be conducted face-to-face, mailed, or even posted on the Internet to get respondents from anywhere in the world.

4. Focus Groups

A focus group is similar to an interview, but it is conducted with a group of people who all have something in common. The data collected is similar to in-person interviews, but they offer a better understanding of why a certain group of people thinks in a particular way.

5. Oral Histories

Oral histories also involve asking questions like interviews and focus groups. However, it is defined more precisely and the data collected is linked to a single phenomenon. It involves collecting the opinions and personal experiences of people in a particular event that they were involved in.

Secondary Data Collection Methods

Secondary data refers to data that has already been collected by someone else. It is much more inexpensive and easier to collect than primary data.

Here are some of the most common secondary data collection methods:

1. Internet

The use of the Internet has become one of the most popular secondary data collection methods in recent times. There is a large pool of free and paid research resources that can be easily accessed on the Internet.

2. Government Archives

There is lots of data available from government archives that you can make use of. The most important advantage is that the data in government archives are authentic and verifiable. The challenge, however, is that data is not always readily available due to a number of factors.

3. Libraries

Most researchers donate several copies of their academic research to libraries. You can collect important and authentic information based on different research contexts.

Data preprocessing

Data preprocessing, a component of data preparation, describes any type of processing performed on raw data to prepare it for another data processing procedure.

Data preprocessing transforms the data into a format that is more easily and effectively processed in data mining, machine learning and other data science tasks. The techniques are generally used at the earliest stages of the machine learning and AI development pipeline to ensure accurate results.

There are several different tools and methods used for preprocessing data, including the following:

- sampling, which selects a representative subset from a large population of data;
- transformation, which manipulates raw data to produce a single input;
- denoising, which removes noise from data;
- imputation, which synthesizes statistically relevant data for missing values;

- normalization, which organizes data for more efficient access; and
- feature extraction, which pulls out a relevant feature subset that is significant in a particular context.

key steps in data preprocessing

1. Data profiling. Data profiling is the process of examining, analyzing and reviewing data to collect statistics about its quality. It starts with a survey of existing data and its characteristics. Data scientists identify data sets that are pertinent to the problem at hand, inventory its significant attributes, and form a hypothesis of features that might be relevant for the proposed analytics or machine learning task.

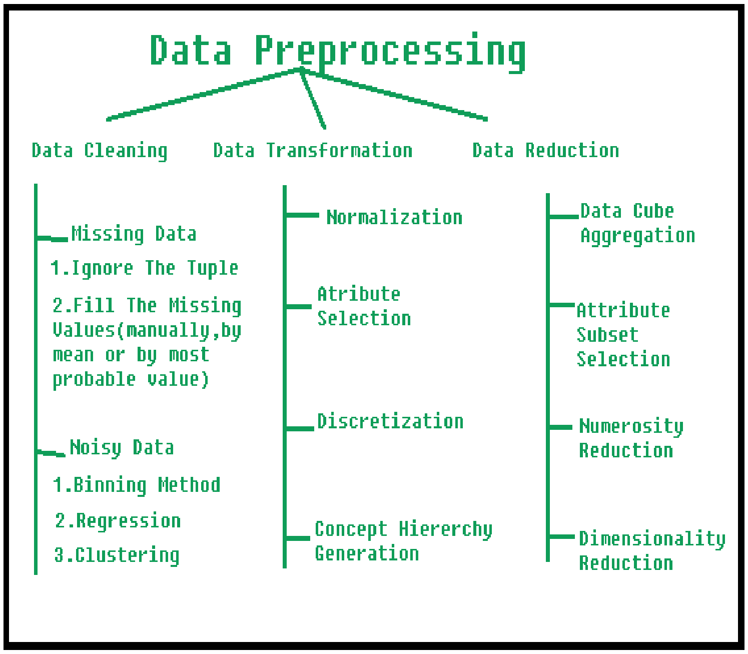
2. Data cleansing. The aim here is to find the easiest way to rectify quality issues, such as eliminating bad data, filling in missing data or otherwise ensuring the raw data is suitable for feature engineering.

3. Data reduction. Raw data sets often include redundant data that arise from characterizing phenomena in different ways or data that is not relevant to a particular ML, AI or analytics task. Data reduction uses techniques like principal component analysis to transform the raw data into a simpler form suitable for particular use cases.

4. Data transformation. Here, data scientists think about how different aspects of the data need to be organized to make the most sense for the goal. This could include things like structuring unstructured data, combining salient variables when it makes sense or identifying important ranges to focus on.

5. Data enrichment. In this step, data scientists apply the various feature engineering libraries to the data to effect the desired transformations. The result should be a data set organized to achieve the optimal balance between the training time for a new model and the required compute.

6. Data validation. At this stage, the data is split into two sets. The first set is used to train a machine learning or deep learning model. The second set is the testing data that is used to gauge the accuracy and robustness of the resulting model.

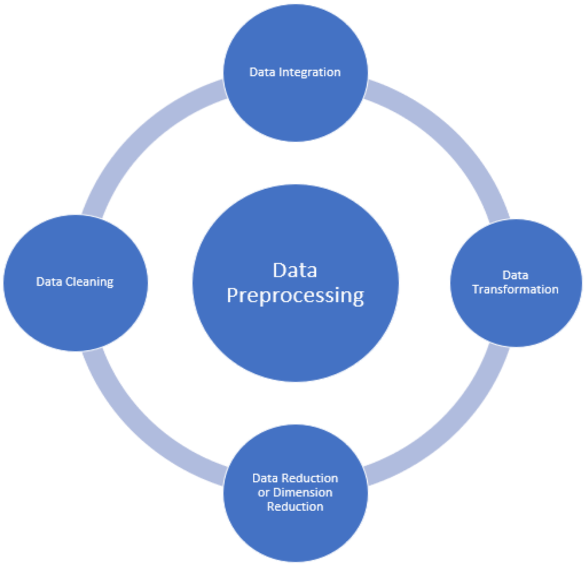


Data Preprocessing in Data Mining

Data preprocessing is the process of transforming raw data into an understandable format. It is also an important step in data mining as we cannot work with raw data. The quality of the data should be checked before applying machine learning or data mining algorithms.

Major Tasks in Data Preprocessing:

1. Data cleaning
2. Data integration
3. Data reduction
4. Data transformation

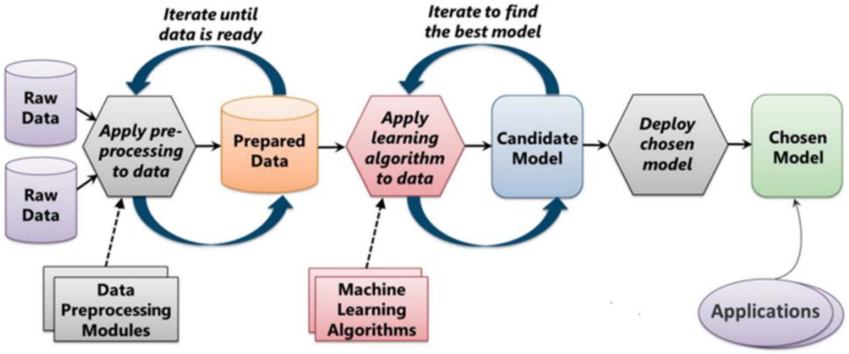


Data Preprocessing in machine learning:

Machine Learning ProcessSteps in Data Preprocessing

- **Step 1 :** Import the libraries
- **Step 2 :** Import the data-set
- **Step 3 :** Check out the missing values
- **Step 4 :** See the Categorical Values
- **Step 5 :** Splitting the data-set into Training and Test Set
- **Step 6 :** Feature Scaling

The Machine Learning Process


Iterate until data is ready
Iterate to find the best model
Raw Data
Raw Data
Apply pre-processing to data
Prepared Data
Apply learning algorithm to data
Candidate Model
Deploy chosen model
Chosen Model
Data Preprocessing Modules
Machine Learning Algorithms
Applications

From "Introduction to Microsoft Azure" by David Chappell

Step 1 : Import the Libraries

This is how we import libraries in Python using import keyword and this is the most popular libraries which any Data Scientist used.

NumPy is the fundamental package for scientific computing with Python. It contains among other things:

1. A powerful N-dimensional array object

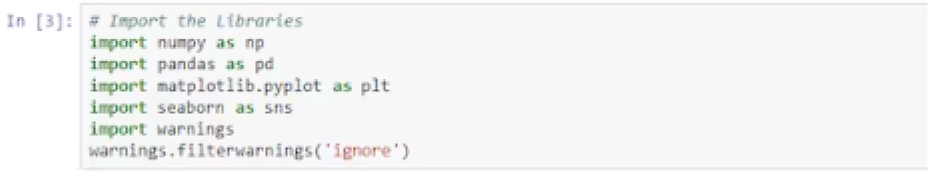
2. Sophisticated (broadcasting) functions

3. Tools for integrating C/C++ and FORTRAN code

4. Useful linear algebra, Fourier transform, and random number capabilities

Pandas *is for* data manipulation and analysis. *Pandas* is an open source, BSD-licensed library providing high-performance, easy-to-use data structures and data analysis tools for the Python programming language.

Matplotlib can be used in Python scripts, the Python and IPython shells, the Jupyter notebook, web application servers, and four graphical user interface toolkits.**Seaborn** is a Python data visualization library based on matplotlib.



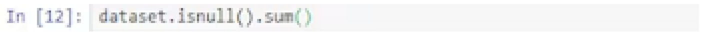
```
In [3]: # Import the Libraries
        import numpy as np
        import pandas as pd
        import matplotlib.pyplot as plt
        import seaborn as sns
        import warnings
        warnings.filterwarnings('ignore')
```

Step 2 : Import the Dataset

By using Pandas we import our data-set and the file I used here is .csv file [Note: It’s not necessarily every-time you deal with **CSV** file, sometimes you deal with **Html or Xlsx(Excel file)**].

Step 3 : Check out the Missing Values

The concept of missing values is important to understand in order to successfully manage data. If the missing values are not handled properly by the researcher, then he/she may end up drawing an inaccurate inference about the data. Due to improper handling, the result obtained by the researcher will differ from ones where the missing values are present.



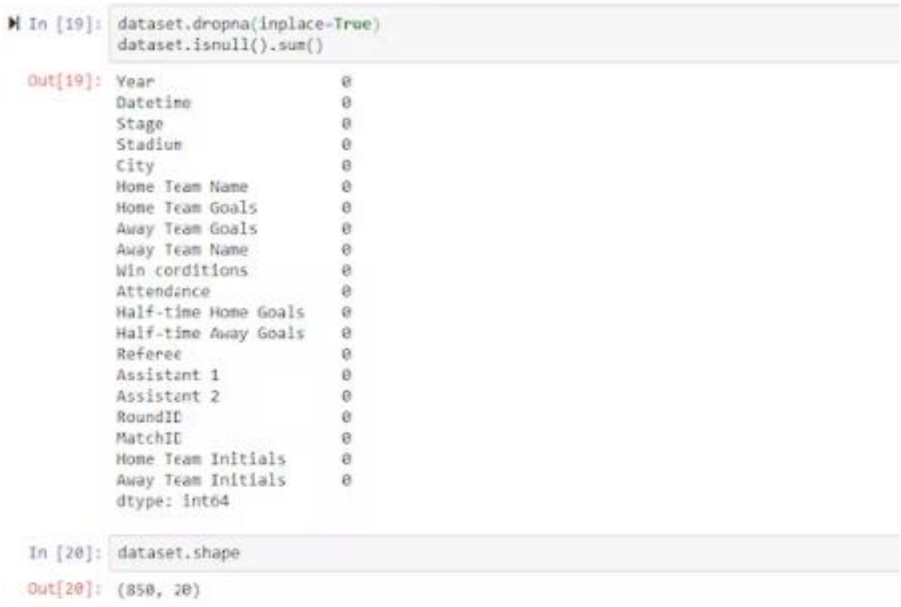
```
In [12]: dataset.isnull().sum()
```

Two ways to handle Missing Values in Data Preprocessing

This data preprocessing method is commonly used to handle the null values.

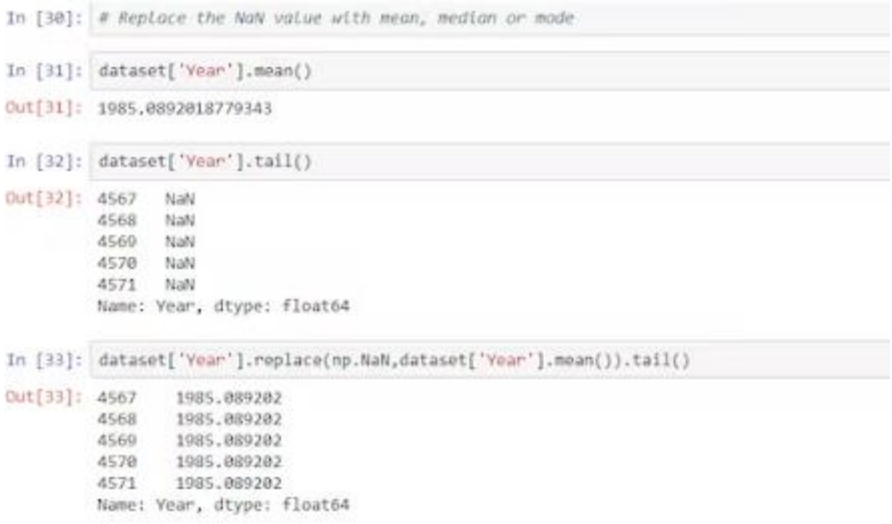
Drop the Missing Values

This strategy can be applied on a feature which has numeric data like the year column or Home team goal column. We can calculate the **mean, median or mode** of the feature and replace it with the missing values.



```
In [19]: dataset.dropna(inplace=True)
         dataset.isnull().sum()

Out[19]: Year                    0
         Datetime                0
         Stage                   0
         Stadium                 0
         City                    0
         Home Team Name          0
         Home Team Goals         0
         Away Team Goals         0
         Away Team Name          0
         Win conditions          0
         Attendance              0
         Half-time Home Goals    0
         Half-time Away Goals    0
         Referee                 0
         Assistant 1             0
         Assistant 2             0
         RoundID                 0
         MatchID                 0
         Home Team Initials      0
         Away Team Initials      0
         dtype: int64

In [20]: dataset.shape

Out[20]: (850, 20)
```

Replace the Missing Value



```
In [30]: # Replace the NaN value with mean, median or mode

In [31]: dataset['Year'].mean()

Out[31]: 1985.0892018779343

In [32]: dataset['Year'].tail()

Out[32]: 4567   NaN
         4568   NaN
         4569   NaN
         4570   NaN
         4571   NaN
         Name: Year, dtype: float64

In [33]: dataset['Year'].replace(np.NaN,dataset['Year'].mean()).tail()

Out[33]: 4567    1985.089202
         4568    1985.089202
         4569    1985.089202
         4570    1985.089202
         4571    1985.089202
         Name: Year, dtype: float64
```

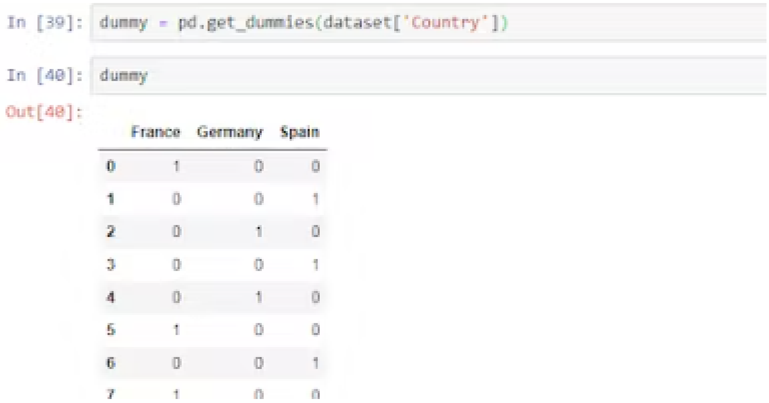
Step 4 : See the Categorical Values

Use LabelEncoder class to convert Categorical data into numerical one

label_encoder is object which is I use and help us in transferring Categorical data into Numerical data. Next, I fitted this label_encoder object to the first column of our matrix X and all this return the first column country of the matrix X encoded.

Dummy Variables is one that takes the value 0 or 1 to indicate the absence or presence of some categorical effect that may be expected to shift the outcome.

Using Pandas to create Dummy Variables



```
In [39]: dummy = pd.get_dummies(dataset['Country'])

In [40]: dummy
Out[40]:
```

	France	Germany	Spain
0	1	0	0
1	0	0	1
2	0	1	0
3	0	0	1
4	0	1	0
5	1	0	0
6	0	0	1
7	1	0	0

Step 5 : Splitting the data-set into Training and Test Set

In any Machine Learning model is that we're going to split data-set into two separate sets

1. Training Set
2. Test Set

Why we need splitting ?

Well here it's your algorithm model that is going to learn from your data to make predictions. Generally we split the data-set into 70:30 ratio or 80:20 what does it mean, 70 percent data take in train and 30 percent data take in test. However, this Splitting can be varies according to the data-set shape and size.

Step 6 : Feature Scaling

Feature scaling is the method to limit the range of variables so that they can be compared on common grounds.

Data cleaning

Data cleaning is fixing or removing incorrect, corrupted, incorrectly formatted, duplicate, or incomplete data within a dataset. If data is incorrect, outcomes and algorithms are unreliable, even though they may look correct. When combining multiple data sources, there are many opportunities for data to be duplicated or mislabeled.

Generally, data cleaning reduces errors and improves data quality. Correcting errors in data and eliminating bad records can be a time-consuming and tedious process, but it cannot be ignored.

Steps of Data Cleaning

1. Remove duplicate or irrelevant observations

Remove unwanted observations from your dataset, including duplicate observations or irrelevant observations. Duplicate observations will happen most often during data collection. When you combine data sets from multiple places, scrape data, or receive data from clients or multiple departments, there are opportunities to create duplicate data. De-duplication is one of the largest areas to be considered in this process.

2. Fix structural errors

Structural errors are when you measure or transfer data and notice strange naming conventions, typos, or incorrect capitalization. These inconsistencies can cause mislabeled categories or classes. For example, you may find "N/A" and "Not Applicable" in any sheet, but they should be analyzed in the same category.

3. **Filter unwanted outliers**

Often, there will be one-off observations where, at a glance, they do not appear to fit within the data you are analyzing. If you have a legitimate reason to remove an outlier, like improper data entry, doing so will help the performance of the data you are working with.

4. Handle missing data

You can't ignore missing data because many algorithms will not accept missing values. There are a couple of ways to deal with missing data. Neither is optimal, but both can be considered, such as:

- You can drop observations with missing values, but this will drop or lose information, so be careful before removing it.

- You can input missing values based on other observations; again, there is an opportunity to lose the integrity of the data because you may be operating from assumptions and not actual observations.
- You might alter how the data is used to navigate null values effectively.

5. Validate and QA

At the end of the data cleaning process, you should be able to answer these questions as a part of basic validation, such as:

- Does the data make sense?
- Does the data follow the appropriate rules for its field?
- Does it prove or disprove your working theory or bring any insight to light?
- Can you find trends in the data to help you for your next theory?

Methods of Data Cleaning

There are many data cleaning methods through which the data should be run. The methods are described below:

1. **Ignore the tuples:** This method is not very feasible, as it only comes to use when the tuple has several attributes is has missing values.
2. **Fill the missing value:** This approach is also not very effective or feasible. Moreover, it can be a time-consuming method. In the approach, one has to fill in the missing value. This is usually done manually, but it can also be done by attribute mean or using the most probable value.
3. **Binning method:** This approach is very simple to understand. The smoothing of sorted data is done using the values around it. The data is then divided into several segments of equal size. After that, the different methods are executed to complete the task.
4. **Regression:** The data is made smooth with the help of using the regression function. The regression can be linear or multiple. Linear regression has only one independent variable, and multiple regressions have more than one independent variable.
5. **Clustering:** This method mainly operates on the group. Clustering groups the data in a cluster. Then, the outliers are detected with the help of clustering. Next, the similar values are then arranged into a "group" or a "cluster".

Process of Data Cleaning

The following steps show the process of data cleaning in data mining.

1. **Monitoring the errors:** Keep a note of suitability where the most mistakes arise. It will make it easier to determine and stabilize false or corrupt information. Information is especially necessary while integrating another possible alternative with established management software.
2. **Standardize the mining process:** Standardize the point of insertion to assist and reduce the chances of duplicity.
3. **Validate data accuracy:** Analyze and invest in data tools to clean the record in real-time. Tools used Artificial Intelligence to better examine for correctness.
4. **Scrub for duplicate data:** Determine duplicates to save time when analyzing data. Frequently attempted the same data can be avoided by analyzing and investing in separate data erasing tools that can analyze rough data in quantity and automate the operation.
5. **Research on data:** Before this activity, our data must be standardized, validated, and scrubbed for duplicates. There are many third-party sources, and these Approved & authorized parties sources can capture information directly from our databases. They help us to clean and compile the data to ensure completeness, accuracy, and reliability for business decision-making.
6. **Communicate with the team:** Keeping the group in the loop will assist in developing and strengthening the client and sending more targeted data to prospective customers.

Tools for Data Cleaning in Data Mining

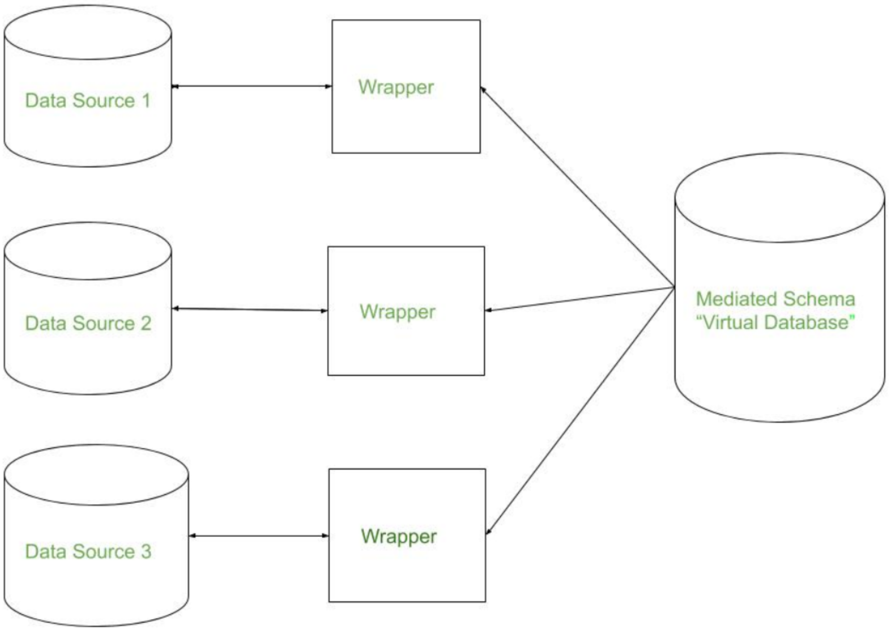
1. OpenRefine
2. Trifacta Wrangler
3. Drake
4. Data Ladder
5. Data Cleaner
6. Cloudingo
7. Reifier

8. IBM Infosphere Quality Stage
9. TIBCO Clarity
10. Winpure

Data Integration

Data Integration is a data preprocessing technique that combines data from multiple heterogeneous data sources into a coherent data store and provides a unified view of the data. These sources may include multiple data cubes, databases, or flat files.

The data integration approaches are formally defined as triple <G, S, M> where,
G stand for the global schema,
S stands for the heterogeneous source of schema,
M stands for mapping between the queries of source and global schema.



There are mainly 2 major approaches for data integration – one is the "tight coupling approach" and another is the "loose coupling approach".

Tight Coupling:

- Here, a data warehouse is treated as an information retrieval component.
- In this coupling, data is combined from different sources into a single physical location through the process of ETL – Extraction, Transformation, and Loading.

Loose Coupling:

- Here, an interface is provided that takes the query from the user, transforms it in a way the source database can understand, and then sends the query directly to the source databases to obtain the result.
- And the data only remains in the actual source databases.

Issues in Data Integration:
There are three issues to consider during data integration: Schema Integration, Redundancy Detection, and resolution of data value conflicts. These are explained in brief below.

1. Schema Integration:
- Integrate metadata from different sources.
- The real-world entities from multiple sources are referred to as the entity identification problem.

2. Redundancy:
- An attribute may be redundant if it can be derived or obtained from another attribute or set of attributes.
- Inconsistencies in attributes can also cause redundancies in the resulting data set.
- Some redundancies can be detected by correlation analysis.

3. Detection and resolution of data value **conflicts:**
- This is the third critical issue in data integration.
- Attribute values from different sources may differ for the same real-world entity.
- An attribute in one system may be recorded at a lower level of abstraction than the "same" attribute in another.

Data Transformation

Data transformation is a technique used to **convert** the raw data into a suitable format that efficiently eases data mining and retrieves strategic information. Data transformation includes data cleaning techniques and a data reduction technique to convert the data into the appropriate form. Data transformation changes the format, structure, or values of the data and converts them into clean, usable data

. Data may be transformed at two stages of the data pipeline for data analytics projects. Organizations that use on-premises data warehouses generally use an ETL (extract, transform, and load) process, in which data transformation is the middle step.

Data Transformation Techniques

1. Data smoothing
2. Attribute construction

3. Data aggregation
4. Data normalization
5. Data discretization
6. Data generalization

1. Data Smoothing

Data smoothing is a process that is used to remove noise from the dataset using some algorithms.

We have seen how the noise is removed from the data using the techniques such as binning, regression, clustering.

- **Binning:** This method splits the sorted data into the number of bins and smoothens the data values in each bin considering the neighborhood values around it.
- **Regression:** This method identifies the relation among two dependent attributes so that if we have one attribute, it can be used to predict the other attribute.
- **Clustering:** This method groups similar data values and form a cluster. The values that lie outside a cluster are known as outliers.

2. Attribute Construction

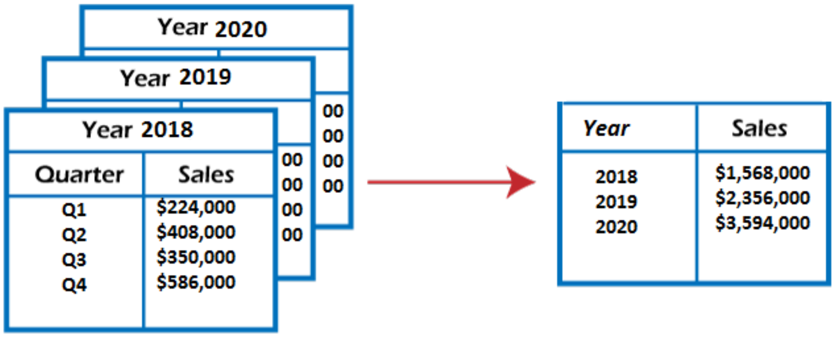
- In the attribute construction method, the new attributes consult the existing attributes to construct a new data set that eases data mining. New attributes are created and applied to assist the mining process from the given attributes. This simplifies the original data and makes the mining more efficient.

Ex: we may have the height and width of each plot. So here, we can construct a new attribute 'area' from attributes 'height' and 'weight'.

3. Data Aggregation

Data collection or aggregation is the method of storing and presenting data in a summary format. The data may be obtained from multiple data sources to integrate these data sources into a data analysis description.

For example, we have a data set of sales reports of an enterprise that has quarterly sales of each year. We can aggregate the data to get the enterprise's annual sales report.



Quarter	Sales
Q1	$224,000
Q2	$408,000
Q3	$350,000
Q4	$586,000

Year	Sales
2018	$1,568,000
2019	$2,356,000
2020	$3,594,000

Aggregated Data

4. Data Normalization

Normalizing the data refers to scaling the data values to a much smaller range such as [-1, 1] or [0.0, 1.0]. There are different methods to normalize the data, as discussed below.

Consider that we have a numeric attribute A and we have n number of observed values for attribute A that are V1, V_2, V_3, ….V_n.

Min-max normalization: This method implements a linear transformation on the original data. Let us consider that we have min$_A$ and max$_A$ as the minimum and maximum value observed for attribute A and $V_i$is the value for attribute A that has to be normalized. The min-max normalization would map V_i to the V'_i in a new smaller range [new_min$_A$, new_max$_A$]. The formula for min-max normalization is given below:

$$v'_i = \frac{v_i - min_A}{max_A - min\ _A}\left(new_{max_A} - new_{min_A}\right) + new\_min_A$$

For example, we have $1200 and $9800 as the minimum, and maximum value for the attribute income, and [0.0, 1.0] is the range in which we have to map a value of $73,600. The value $73,600 would be transformed using min-max normalization as follows:

$$\frac{73600-1200}{9800-1200}(1.0 - 0.0) + 0.0 = 0.716$$

Z-score normalization: This method normalizes the value for attribute A using the ***mean***and ***standard deviation***. The following formula is used for Z-score normalization:

$$v'_i = \frac{v_i - \bar{A}}{\sigma_A}$$

Here Ā and σ_A are the mean and standard deviation for attribute A, respectively. For example, we have a mean and standard deviation for attribute A as $54,000 and $16,000. And we have to normalize the value $73,600 using z-score normalization.

$$\frac{73600-5400}{1600} = 1.225$$

Decimal Scaling: This method normalizes the value of attribute A by moving the decimal point in the value. This movement of a decimal point depends on the maximum absolute value of A. The formula for the decimal scaling is given below:

$$v'_i = \frac{v_i}{10^j}$$

Here j is the smallest integer such that max(|v'_i|)<1
For example, the observed values for attribute A range from -986 to 917, and the maximum absolute value for attribute A is 986. Here, to normalize each value of attribute A using decimal scaling, we have to divide each value of attribute A by 1000, i.e., j=3.
So, the value -986 would be normalized to -0.986, and 917 would be normalized to 0.917.

5. Data Discretization

This is a process of converting continuous data into a set of data intervals. Continuous attribute values are substituted by small interval labels. This makes the data easier to study and analyze.

Data discretization can be classified into two types: **supervised discretization**, where the class information is used, and **unsupervised discretization**, which is based on which direction the process proceeds, i.e., 'top-down splitting strategy' or 'bottom-up merging strategy'.

For example, the values for the age attribute can be replaced by the interval labels such as (0-10, 11-20…) or (kid, youth, adult, senior).

6. Data Generalization

It converts low-level data attributes to high-level data attributes using concept hierarchy. This conversion from a lower level to a higher conceptual level is useful to get a clearer picture of the data. Data generalization can be divided into two approaches:

- Data cube process (OLAP) approach.
- Attribute-oriented induction (AOI) approach.

For example, age data can be in the form of (20, 30) in a dataset. It is transformed into a higher conceptual level into a categorical value (young, old).

Data Reduction

Data reduction techniques ensure the integrity of data while reducing the data. Data reduction is a process that reduces the volume of original data and represents it in a much smaller volume.

Techniques of Data Reduction

1. Dimensionality reduction
2. Numerosity reduction
3. Data cube aggregation
4. Data compression
5. Discritization operation

1. Dimensionality Reduction

Whenever we encounter weakly important data, we use the attribute required for our analysis. Dimensionality reduction eliminates the attributes from the data set under consideration, thereby reducing the volume of original data.

Here are three methods of dimensionality reduction.

- **Wavelet Transform:** In the wavelet transform, suppose a data vector A is transformed into a numerically different data vector A' such that both A and A' vectors are of the same length.
- **Principal Component Analysis:** Suppose we have a data set to be analyzed that has tuples with n attributes. The principal component analysis identifies k independent tuples with n attributes that can represent the data set.
- **Attribute Subset Selection:** The large data set has many attributes, some of which are irrelevant to data mining or some are redundant. The core attribute subset selection reduces the data volume and dimensionality.

2.Numerosity Reduction

The numerosity reduction reduces the original data volume and represents it in a much smaller form. This technique includes two types parametric and non-parametric numerosity reduction.

i) Parametric: Parametric numerosity reduction incorporates storing only data parameters instead of the original data. One method of parametric numerosity reduction is the regression and log-linear method.

- **Regression and Log-Linear:** Linear regression models a relationship between the two attributes by modeling a linear equation to the data set. Suppose we need to model a linear function between two attributes.

  y=wx+b

  Here, y is the response attribute, and x is the predictor attribute.

Log-linear model discovers the relation between two or more discrete attributes in the database. Suppose we have a set of tuples presented in n-dimensional space. Then the log-linear model is used to study the probability of each tuple in a multidimensional space.

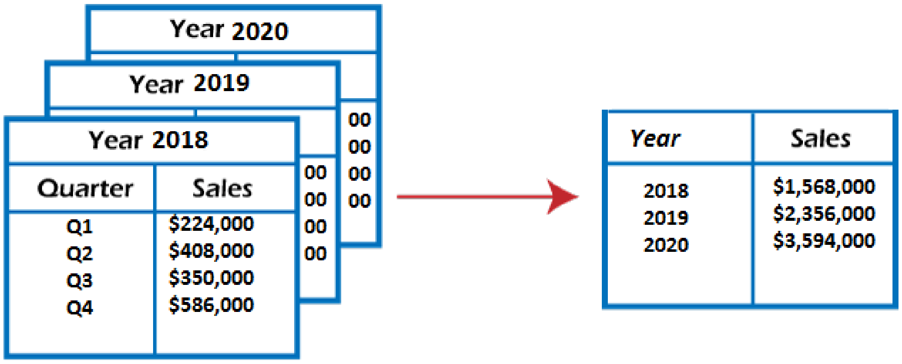
ii) Non-Parametric: A non-parametric numerosity reduction technique does not assume any model. The non-Parametric technique results in a more uniform reduction, irrespective of data size, but it may not achieve a high volume of data reduction like the parametric.

Non-Parametric data reduction techniques, Histogram, Clustering, Sampling, Data Cube Aggregation, and Data Compression.

- **Histogram:** A histogram is a graph that represents frequency distribution which describes how often a value appears in the data. Histogram uses the binning method to represent an attribute's data distribution. It uses a disjoint subset which we call bin or buckets.
- **Clustering:** Clustering techniques groups similar objects from the data so that the objects in a cluster are similar to each other, but they are dissimilar to objects in another cluster.
- **Sampling:** One of the methods used for data reduction is sampling, as it can reduce the large data set into a much smaller data sample.
- **Cluster sample:** The tuples in data set D are clustered into M mutually disjoint subsets. The data reduction can be applied by implementing SRSWOR on these clusters. A simple random sample of size s could be generated from these clusters where s<M.
- **Stratified sample:** The large data set D is partitioned into mutually disjoint sets called 'strata'. A simple random sample is taken from each stratum to get stratified data. This method is effective for skewed data.

3. Data Cube Aggregation

This technique is used to aggregate data in a simpler form. Data Cube Aggregation is a multidimensional aggregation that uses aggregation at various levels of a data cube to represent the original data set, thus achieving data reduction.



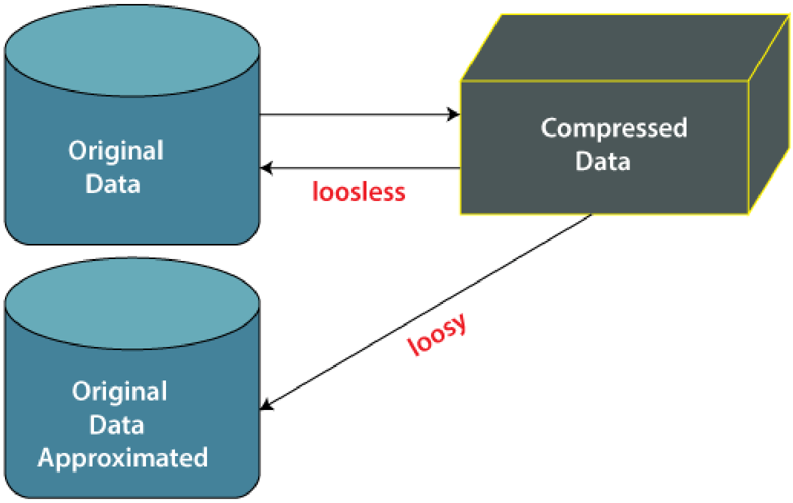
Year 2018	
Quarter	Sales
Q1	$224,000
Q2	$408,000
Q3	$350,000
Q4	$586,000

Year	Sales
2018	$1,568,000
2019	$2,356,000
2020	$3,594,000

Aggregated Data

4. Data Compression

Data compression employs modification, encoding, or converting the structure of data in a way that consumes less space. Data compression involves building a compact representation of information by removing redundancy and representing data in binary form. Data that can be restored successfully from its compressed form is called Lossless compression.



i. **Lossless Compression:** Encoding techniques (Run Length Encoding) allow a simple and minimal data size reduction. Lossless data compression uses algorithms to restore the precise original data from the compressed data.

ii. **Lossy Compression:** In lossy-data compression, the decompressed data may differ from the original data but are useful enough to retrieve information from them. For example, the

JPEG image format is a lossy compression, but we can find the meaning equivalent to the original image. Methods such as the Discrete Wavelet transform technique PCA (principal component analysis) are examples of this compression.

5. Discretization Operation

The data discretization technique is used to divide the attributes of the continuous nature into data with intervals. We replace many constant values of the attributes with labels of small intervals. This means that mining results are shown in a concise and easily understandable way.

i. **Top-down discretization:** If you first consider one or a couple of points (so-called breakpoints or split points) to divide the whole set of attributes and repeat this method up to the end, then the process is known as top-down discretization, also known as splitting.

ii. **Bottom-up discretization:** If you first consider all the constant values as split-points, some are discarded through a combination of the neighborhood values in the interval. That process is called bottom-up discretization.

Data Discretization

Data discretization refers to a method of converting a huge number of data values into smaller ones so that the evaluation and management of data become easy. In other words, data discretization is a method of converting attributes values of continuous data into a finite set of intervals with minimum data loss.

There are two forms of data discretization first is supervised discretization, and the second is unsupervised discretization.

Supervised discretization refers to a method in which the class data is used. Unsupervised discretization refers to a method depending upon the way which operation proceeds. It means it works on the top-down splitting strategy and bottom-up merging strategy.

Now, we can understand this concept with the help of an example

Suppose we have an attribute of Age with the given values

Age	1,5,9,4,7,11,14,17,13,18, 19,31,33,36,42,44,46,70,74,78,77

Table before Discretization

Attribute	Age	Age	Age	Age
	1,5,4,9,7	11,14,17,13,18,19	31,33,36,42,44,46	70,74,77,78
After Discretization	Child	Young	Mature	Old

Some Famous techniques of data discretization

Histogram analysis

Histogram refers to a plot used to represent the underlying frequency distribution of a continuous data set. Histogram assists the data inspection for data distribution. For example, Outliers, skewness representation, normal distribution representation, etc.

Binning

Binning refers to a data smoothing technique that helps to group a huge number of continuous values into smaller values. For data discretization and the development of idea hierarchy, this technique can also be used.

Cluster Analysis

Cluster analysis is a form of data discretization. A clustering algorithm is executed by dividing the values of x numbers into clusters to isolate a computational feature of x.

Data discretization using decision tree analysis

data discretization refers to a decision tree analysis in which a top-down slicing technique is used. It is done through a supervised procedure. In a numeric attribute discretization, first, you need to select the attribute that has the least entropy, and then you need to run it with the help of a recursive process.

Data discretization and concept hierarchy generation

The term hierarchy represents an organizational structure or mapping in which items are ranked according to their levels of importance. In other words, we can say that a hierarchy concept refers to a sequence of mappings with a set of more general concepts to complex concepts. It means mapping is done from low-level concepts to high-level concepts.

Top-down mapping

Top-down mapping generally starts with the top with some general information and ends with the bottom to the specialized information.

Bottom-up mapping

Bottom-up mapping generally starts with the bottom with some specialized information and ends with the top to the generalized information.

UNIT 3

DESCRIPTIVE STATISTICS

Descriptive statistics:
Descriptive statistics describe, show, and summarize the basic features of a dataset found in a given study, presented in a summary that describes the data sample and its measurements. It helps analysts to understand the data better.

Descriptive statistics represent the available data sample and does not include theories, inferences, probabilities, or conclusions. That’s a job for inferential statistics.

If you want a good example of descriptive statistics, look no further than a student’s grade point average (GPA). A GPA gathers the data points created through a large selection of grades, classes, and exams, then averages them together and presents a general idea of the student’s mean academic performance. Note that the GPA doesn’t predict future performance or present any conclusions.

Types of Descriptive Statistics

Descriptive statistics break down into several types, characteristics, or measures. Some authors say that there are two types. Others say three or even four. In the spirit of working with averages, we will go with three types.

- Distribution, which deals with each value’s frequency
- Central tendency, which covers the averages of the values
- Variability (or dispersion), which shows how spread out the values are

Distribution (also called Frequency Distribution)

Datasets consist of a distribution of scores or values. Statisticians use graphs and tables to summarize the frequency of every possible value of a variable, rendered in percentages or numbers. For instance, if you held a poll to determine people’s favorite Beatle, you’d set up one column with all possible variables (John, Paul, George, and Ringo), and another with the number of votes.

Statisticians depict frequency distributions as either a graph or as a table.

Measures of Central Tendency

Measures of central tendency estimate a dataset's average or center, finding the result using three methods: mean, mode, and median.

Mean. The mean is also known as “M” and is the most common method for finding averages. You get the mean by adding all the response values together, dividing the sum by the number of responses, or “N.” For instance, say someone is trying to figure out how many hours a day they sleep in a week. So, the data set would be the hour entries (e.g., 6,8,7,10,8,4,9), and the sum of those values is 52. There are seven responses, so N=7. You divide the value sum of 52 by N, or 7, to find M, which in this instance is 7.3.

Mode. The mode is just the most frequent response value. Datasets may have any number of modes, including “zero.” You can find the mode by arranging your dataset's order from the lowest to highest value and then looking for the most common response. So, in using our sleep study from the last part: 4,6,7,8,8,9,10. As you can see, the mode is eight.

Median. Finally, we have the median, defined as the value in the precise center of the dataset. Arrange the values in ascending order (like we did for the mode) and look for the number in the set's middle. In this case, the median is eight.

Variability (also called Dispersion)

The measure of variability gives the statistician an idea of how spread out the responses are. The spread has three aspects — range, standard deviation, and variance.

Range. Use range to determine how far apart the most extreme values are. Start by subtracting the dataset's lowest value from its highest value. Once again, we turn to our sleep study: 4,6,7,8,8,9,10. We subtract four (the lowest) from ten (the highest) and get six. There's your range.

Standard Deviation. This aspect takes a little more work. The standard deviation (s) is your dataset's average amount of variability, showing you how far each score lies from the mean. The larger your standard deviation, the greater your dataset's variable. Follow these six steps:

1. List the scores and their means.
2. Find the deviation by subtracting the mean from each score.
3. Square each deviation.
4. Total up all the squared deviations.
5. Divide the sum of the squared deviations by N-1.
6. Find the result's square root.

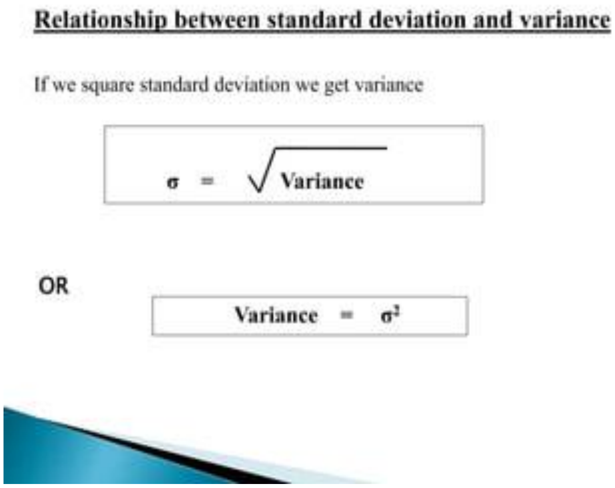
Example: we turn to our sleep study: 4,6,7,8,8,9,10.

Raw Number/Data	Deviation from Mean	Deviation Squared
4	4-7.3= -3.3	10.89

6	6-7.3= -1.3	1.69
7	7-7.3= -0.3	0.09
8	8-7.3= 0.7	0.49
8	8-7.3= 0.7	0.49
9	9-7.3=1.7	2.89
10	10-7.3= 2.7	7.29
M=7.3	Sum = 0.9	Square sums= 23.83

When you divide the sum of the squared deviations by 6 (N-1): 23.83/6, you get 3.971, and the square root of that result is 1.992. As a result, we now know that each score deviates from the mean by an average of 1.992 points.

Variance: Variance reflects the dataset's degree spread. The greater the degree of data spread, the larger the variance relative to the mean. You can get the variance by just squaring the standard deviation. Using our above example, we square 1.992 and arrive at 3.971.



Relationship between standard deviation and variance

If we square standard deviation we get variance

$$\sigma = \sqrt{\text{Variance}}$$

OR

$$\text{Variance} = \sigma^2$$

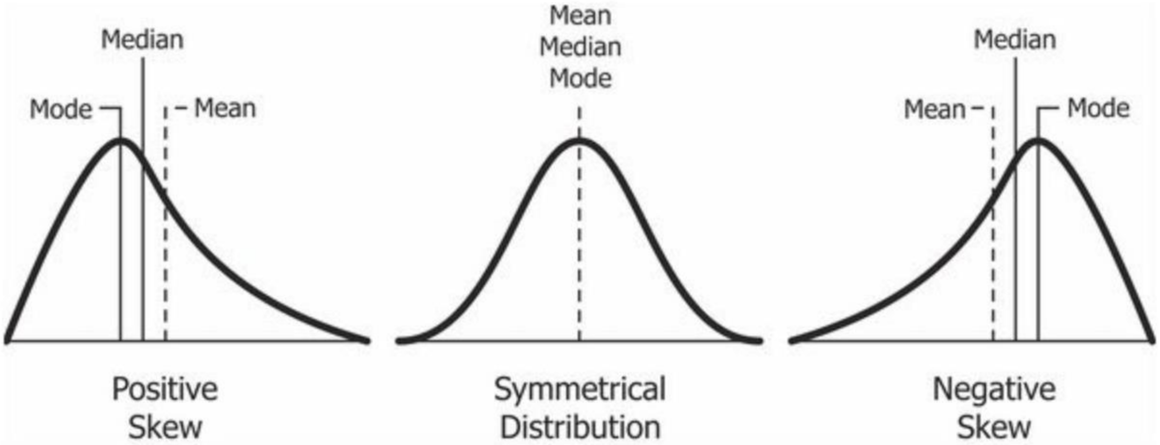
Skewness:

Skewness is the measure of the asymmetry of an ideally symmetric probability distribution and is given by the third standardized moment. If that sounds way too complex, don’t worry! Let me break it down for you.

In simple words, skewness is the measure of how much the probability distribution of a random variable deviates from the normal distribution.

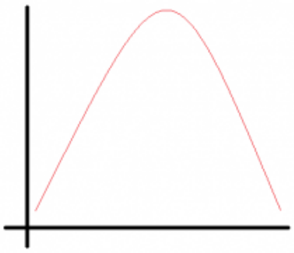
the normal distribution is the probability distribution without any skewness. You can look at the image below which shows symmetrical distribution that’s basically a normal distribution and you can see that it is symmetrical on both sides of the dashed line. Apart from this, there are two types of skewness:

- Positive Skewness
- Negative Skewness



The probability distribution with its tail on the right side is a positively skewed distribution and the one with its tail on the left side is a negatively skewed distribution.

- **Normal Distribution:** Normal Distribution is a probability distribution that is symmetric about the mean. It is also known as Gaussian Distribution. The distribution appears as a Bell-shaped curve which means the mean is the most frequent data in the given data set.



In Normal Distribution :

Mean = Median = Mode

- **Standard Normal Distribution:** When in the Normal Distribution **mean = 0** and the Standard Deviation = 1 then Normal Distribution is called as Standard Normal Distribution.
- Normal Distributions are symmetrical in nature it doesn’t imply that every symmetrical distribution is a Normal Distribution.
- Normal Distribution is the probability distribution without any skewness.

Types of Skewness

- Positive Skewness
- Negative Skewness

Unlike the Normal Distribution (mean = median = mode), in positive as well as negative skewness mean, median, and mode all are different.

Positive Skewness

In positive skewness, the extreme data values are larger, which in turn increase the mean value of the data set, or in the simple term in positive skew distribution is the distribution having the tail on the right side.

In Positive Skewness:

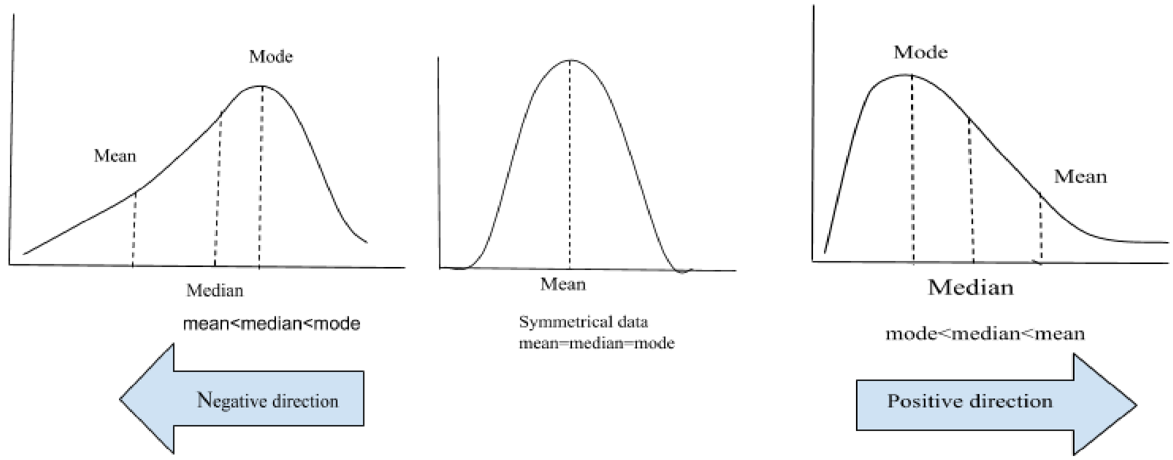
Mean > Median > Mode

Negative Skewness

In negative skewness, the extreme data values are smaller, which decreases the mean value of the dataset or the negative skew distribution is the distribution having the tail on the left side.

In Negative Skewness:

Mean < Median< Mode



Calculate the skewness coefficient of the sample

Pearson's first coefficient of skewness

Subtract a mode from a mean, then divides the difference by standard deviation.

$$\textbf{Pearson's first coefficient} = \frac{\textbf{Mean – Mode}}{\textbf{Standard Deviation}}$$

As Pearson's correlation coefficient differs from -1 (perfect negative linear relationship) to +1 (perfect positive linear relationship), including a value of 0 indicating no linear relationship, When we divide the covariance values by the standard deviation, it truly scales the value down to a limited range of **-1 to +1.** That accurately the range of the correlation values.

Pearson's second coefficient of skewness

Multiply the difference by 3, and divide the product by standard deviation.

$$\text{Pearson's second coefficient} = \frac{3\,(\text{Mean} - \text{Median})}{\text{Standard Deviation}}$$

$$\text{Mean} - \text{Mode} \approx 3\,(\text{Mean} - \text{Median})$$

If the skewness is between -0.5 & 0.5, the data are nearly symmetrical.

If the skewness is between -1 & -0.5 (negative skewed) or between 0.5 & 1(positive skewed), the data are slightly skewed.

If the skewness is lower than -1 (negative skewed) or greater than 1 (positive skewed), the data are extremely skewed.

Kurtosis:

Kurtosis measures the "heaviness of the tails" of a distribution (in compared to a normal distribution). Kurtosis is positive if the tails are "heavier" then for a normal distribution, and negative if the tails are "lighter" than for a normal distribution. The normal distribution has kurtosis of zero.

Kurtosis characterizes the shape of a distribution - that is, its value does not depend on an arbitrary change of the scale and location of the distribution. For example, kurtosis of a sample (or population) of temperature values in Fahrenheit will not change if you transform the values to Celsius (the mean and the variance will, however, change).

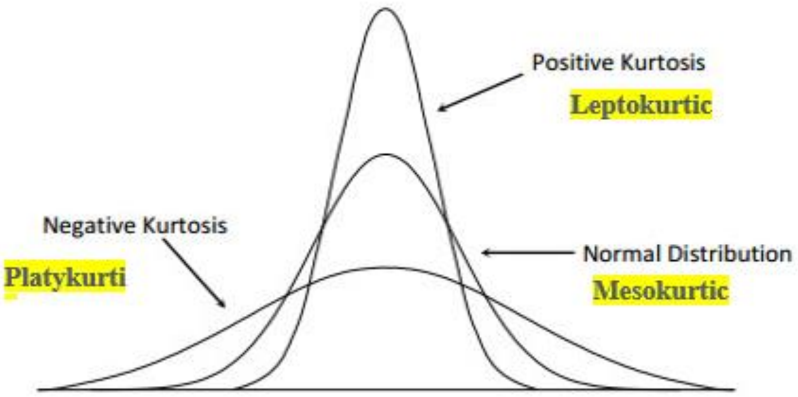
The kurtosis of a distribution or sample is equal to the 4th central moment divided by the 4th power of the standard deviation, minus 3.

To calculate the kurtosis of a sample:

i) subtract the mean from each value to get a set of deviations from the mean;

ii) divide each deviation by the standard deviation of all the deviations;

iii) average the 4th power of the deviations and subtract 3 from the result.



Excess Kurtosis

The excess kurtosis is used in statistics and probability theory to compare the kurtosis coefficient with that normal distribution. Excess kurtosis can be positive (Leptokurtic distribution), negative (Platykurtic distribution), or near to zero (Mesokurtic distribution). Since normal distributions have a kurtosis of 3, excess kurtosis is calculating by subtracting kurtosis by 3.

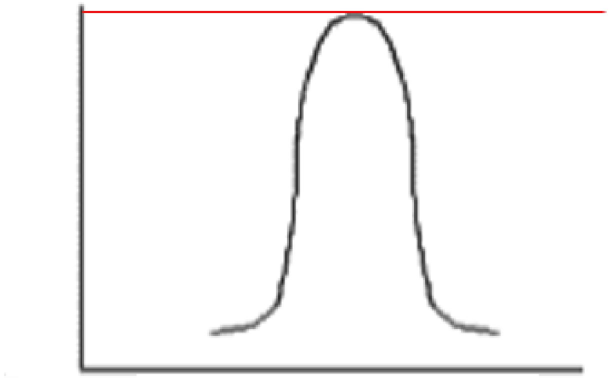
Excess kurtosis = Kurt – 3

Types of excess kurtosis

1. *Leptokurtic or heavy-tailed distribution (kurtosis more than normal distribution).*
2. *Mesokurtic (kurtosis same as the normal distribution).*
3. *Platykurtic or short-tailed distribution (kurtosis less than normal distribution).*

Leptokurtic (kurtosis > 3)

Leptokurtic is having very long and skinny tails, which means there are more chances of outliers. Positive values of kurtosis indicate that distribution is peaked and possesses thick tails. An extreme positive kurtosis indicates a distribution where more of the numbers are located in the tails of the distribution instead of around the mean.



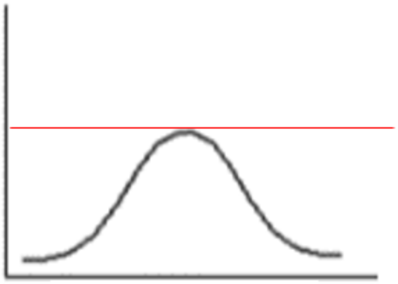
Leptokurtic

platykurtic (kurtosis < 3)

Platykurtic having a lower tail and stretched around center tails means most of the data points are present in high proximity with mean. A platykurtic distribution is flatter (less peaked) when compared with the normal distribution.

Mesokurtic (kurtosis = 3)

Mesokurtic is the same as the normal distribution, which means kurtosis is near to 0. In Mesokurtic, distributions are moderate in breadth, and curves are a medium peaked height.



Mesokurtic

Mesokurtic = 3 – 3 = 0

Box plot:

A box plot also known as Five Number Summary, summarizes data using the median, upper quartile, lower quartile, and the minimum and maximum values. It allows you to see important characteristics of the data at a glance(visually). This also help us to visualize outliers in the data set.

Box plot or Five Number Summary has below five information.

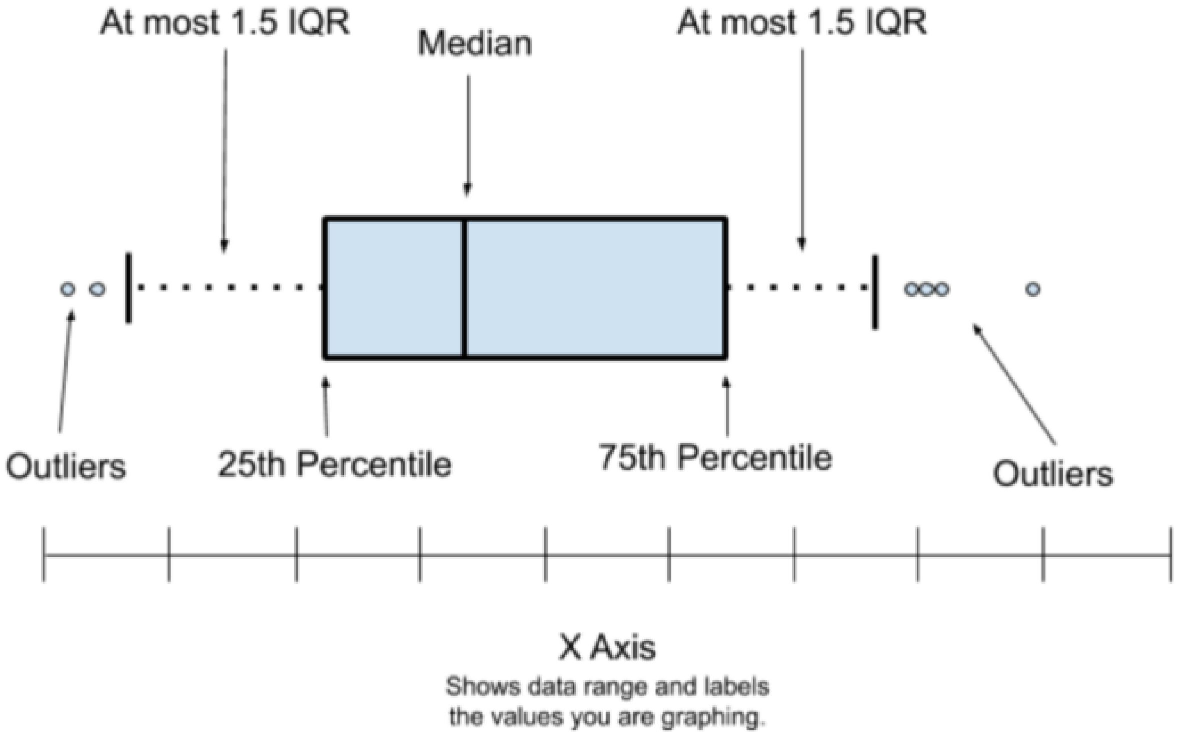
1.Median

2. Lower Quartile(25th Percentile)

3.Upper Quartile(75th Percentile)

4. Minimum Value

5.Maximum Value



Working Example of Box

Let’s understand Box plot with this an example.

Step 1 — take the set of numbers given

14, 19, 100, 27, 54, 52, 93, 50, 61, 87,68, 85, 75, 82, 95

Arrange the data in increasing(ascending) order

14, 19, 27, 50, 52, 54, 61, 68, 75, 82, 85, 87, 93, 95, 100

Step 2 — Find the median of this data set. Median is mid value in this ordered data set.

14, 19, 27, 50, 52, 54, 61, **68**, 75, 82, 85, 87, 93, 95, 100

Here it is 68.

Step 3 — Lets find the Lower Quartile.

Lower Quartile is the median from the left of the, medium found in the Step 2(ie. 68)

(14, 19, 27, **50**, 52, 54, 61), 68, 75, 82, 85, 87, 93, 95, 100

Lower Quartile is 50

Step 4 — Lets find the Upper Quartile.

Upper Quartile is the median from the Right of the medium found in the Step 2(ie. 68)

14, 19, 27, 50, 52, 54, 61, 68,(75, 82, 85, **87**, 93, 95, 100)

Upper Quartile is 87

Step 5 — Lets find the Minimum Value

It is value lies in the extreme left from this data set or first value in the data set after ordering.

14, 19, 27, 50, 52, 54, 61, 68, 75, 82, 85, 87, 93, 95, 100

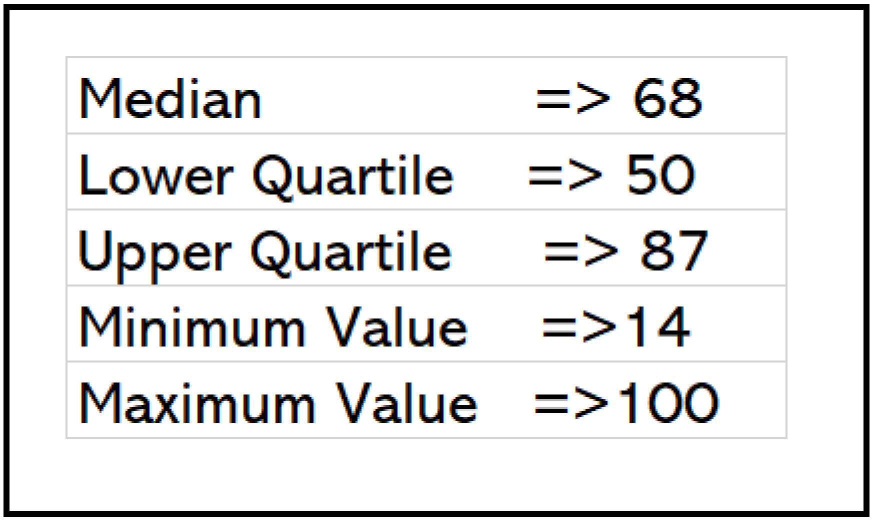
Minimum Value is 14

Step 6 — Lets find the Maximum Value

It is value lies in the extreme Right from this data set or last value in the data set after ordering.

14, 19, 27, 50, 52, 54, 61, 68, 75, 82, 85, 87, 93, 95, **100**

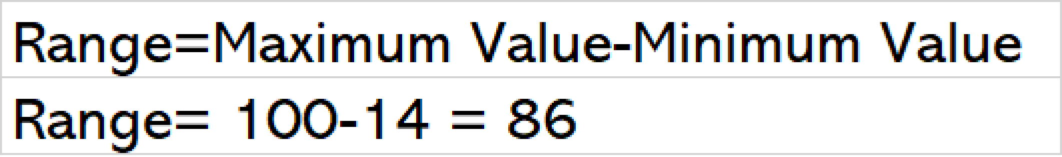
Maximum Value is 100



Median	=> 68
Lower Quartile	=> 50
Upper Quartile	=> 87
Minimum Value	=>14
Maximum Value	=>100

Range :

Range is basically spread of our data set.Range can be found as difference between Maximum Value and Minimum Values.

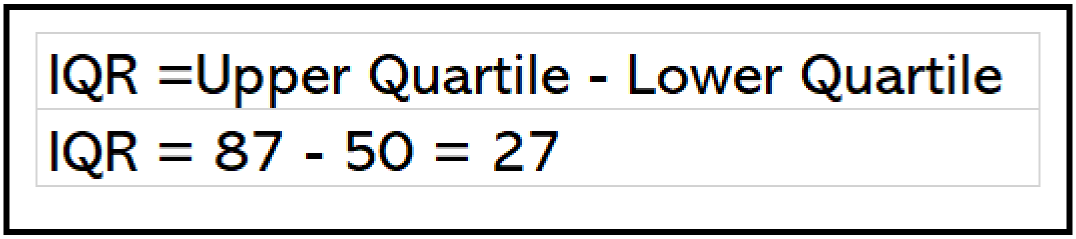


Range=Maximum Value-Minimum Value
Range= 100-14 = 86

Interquartile Range(IQR):

Interquartile Range(IQR) is difference between Upper quartile and Lower quartile.

As per picture above,our Lower quartile is 50 and Upper quartile is 87



IQR =Upper Quartile - Lower Quartile
IQR = 87 - 50 = 27

Box plot with Even numbers of data set :

Step 1 :

We have 14 records below.

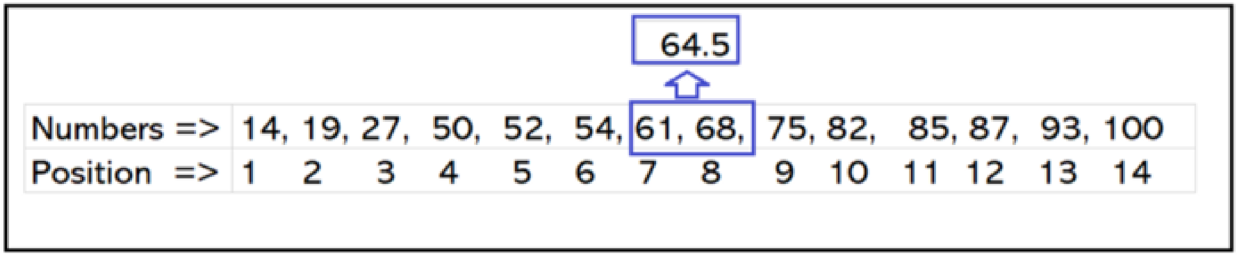
14, 19, 100, 27, 54, 52, 93, 50, 61, 87,68, 85, 75, 82

Arrange the data in increasing(ascending) order

14, 19, 27, 50, 52, 54, 61, 68, 75, 82, 85, 87, 93, 100

Step 2:Since we the even number take the middle two values add them and divide them by 2.

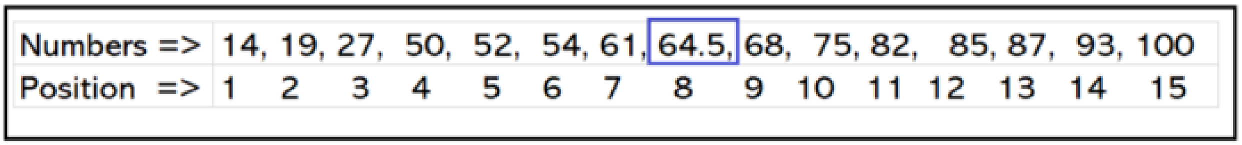
Here Values at position 7 & 8 are middle values.



64.5

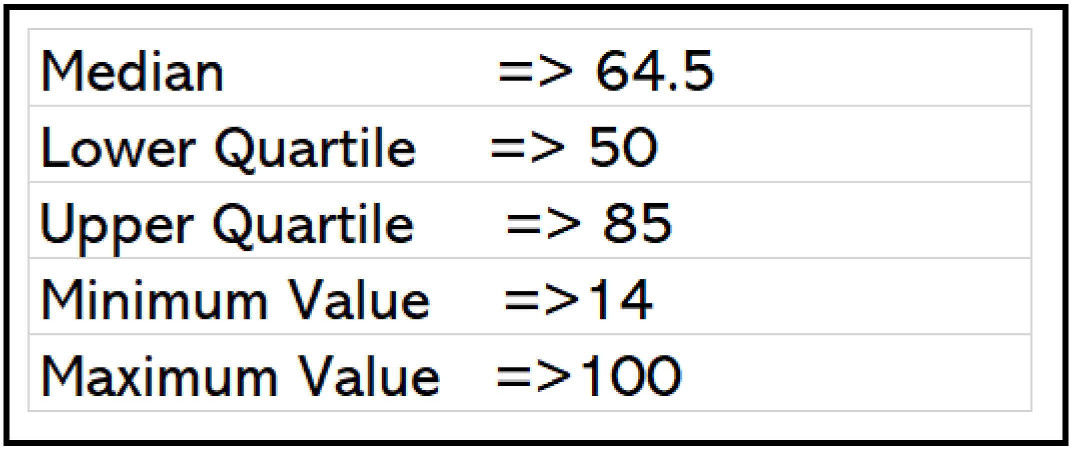
Numbers =>	14,	19,	27,	50,	52,	54,	61,	68,	75,	82,	85,	87,	93,	100
Position =>	1	2	3	4	5	6	7	8	9	10	11	12	13	14

So our new median value is 64.5



Numbers =>	14,	19,	27,	50,	52,	54,	61,	64.5,	68,	75,	82,	85,	87,	93,	100
Position =>	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

Continue Step 3 to Step 6 to get the values mention in **Working Example of Box Plot** section**.**Final Result as below



Median	=> 64.5
Lower Quartile	=> 50
Upper Quartile	=> 85
Minimum Value	=>14
Maximum Value	=>100

Pivot table:

A pivot table is a **summary tool** that wraps up or summarizes information sourced from bigger tables. These bigger tables could be a database, an Excel spreadsheet, or any data that is or could be converted in a table-like form. The data summarized in a pivot table might include sums, averages, or other statistics which the pivot table groups together in a meaningful way.

Wide vs. Long Data

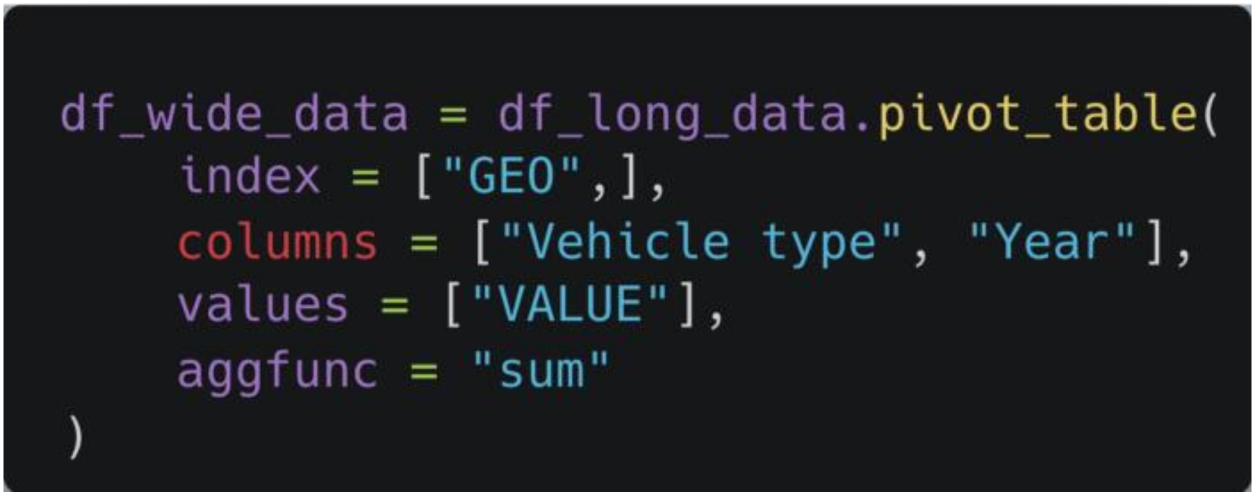
Pivoting is the act of converting “long data” into “wide data”. Wide and long data formats serve different purposes. It’s often helpful to think of how data might be collected in the first place.

Representing potentially multi-dimensional data in a single 2-dimensional table may require some compromises. Either some data will be repeated (long data format) or your data set may require blank cells (wide data).

Examples of pivot table:

- **Pivoting in Python with Pandas**

Starting with the raw dataset loaded into `df_long_data`



```
df_wide_data = df_long_data.pivot_table(
    index = ["GEO",],
    columns = ["Vehicle type", "Year"],
    values = ["VALUE"],
    aggfunc = "sum"
)
```

Note the key arguments here.

1. **Index** determines what will be unique in the leftmost column of the result.
2. **Columns** creates a column for each unique value in the “Vehicle type” and “Year” columns of the input table.
3. **Values** defines what to put in the cells of the output table.
4. **aggfunc** defines *how* to combine the values (commonly sum, but other aggregation functions are also used: min, max, mean, 95th percentile, mode). You can also define your own aggregation functions and pass in the function.

- **Pivoting in Google Sheets**

Pivoting in Google sheets is doing the exact same thing as our python code above but in a graphical way. Instead of specifying arguments for the `pivot_table` function, we select from dropdowns. But the important thing to remember is pivoting doesn’t “belong” to any particular software, it’s a generalized approach for dealing with data.

- **Grouping in PostgreSQL**

Many relational databases don’t have a built in pivot function. We can write a query that approximates the desired results but it does require some manual intervention to define the possible groups.

Aggregation Functions

Sum is often used to combine data in pivot tables but pivot tables are much more flexible than just simple sums. Different tools will provide their own selection of **aggregation functions.** For example, Pandas provides: `min, max, first, last, unique, std (standard deviation),` `var (variance), count, unique, quantile` among others. We can also define our own aggregation functions and pass multiple different aggregation functions to the same column or different columns in the same pivot table.

Missing Data

As with any aggregation missing data must be dealt with. In the example data used here there are numerous months with `0` sales. The most probably explanation is that data wasn't collected for those month, not that there were actually zero sales. There's also the scenario where data is missing entirely and the "cell" is blank or contains a `null` value. Whatever the program you'll need to accept (and be aware of) the default behaviour for these cases or specify what to do.

Heat map:

Heatmaps visualize the data in a 2-dimensional format in the form of colored maps. The color maps use hue, saturation, or luminance to achieve color variation to display various details. This color variation gives visual cues to the readers about the magnitude of numeric values.

Heatmaps can describe the density or intensity of variables, visualize patterns, variance, and even anomalies. Heatmaps show relationships between variables. These variables are plotted on both axes.

Uses of HeatMap

Business Analytics: A heat map is used as a visual business analytics tool. A heat map gives quick visual cues about the current results, performance, and scope for improvements. Heatmaps can analyze the existing data and find areas of intensity that might reflect where most customers reside, areas of risk of market saturation, or cold sites and sites that need a boost. Heat maps can be continued to be updated to reflect the growth and efforts. These maps can be integrated into a business's workflow and become a part of ongoing analytics.

- **Website:** Heatmaps are used in websites to visualize data of visitors' behavior. This visualization helps business owners and marketers to identify the best & worst-performing sections of a webpage. These insights help them with optimization.
    - **Exploratory Data Analysis:** EDA is a task performed by data scientists to get familiar with the data. All the initial studies are done to understand the data are known as EDA. Exploratory Data Analysis (EDA) is the process of analyzing datasets before the modeling task. It is a tedious task to look at a spreadsheet filled with numbers and determine essential characteristics in a dataset.

- **Molecular Biology:** Heat maps are used to study disparity and similarity patterns in DNA, RNA, etc.

    - **Geovisualization:** Geospatial heatmap charts are useful for displaying how geographical areas of a map are compared to one another based on specific criteria. Heatmaps help in cluster analysis or hotspot analysis to detect clusters of high concentrations of activity; For example, Airbnb rental price analysis.

    - **Marketing and Sales:** The heatmap's capability to detect warm and cold spots is used to improve marketing response rates by targeted marketing. Heatmaps allow the detection of areas that respond to campaigns, under-served markets, customer residence, and high sale trends, which helps optimize product lineups, capitalize on sales, create targeted customer segments, and assess regional demographics.

Types of HeatMaps

Typically, there are two types of Heatmaps:

- **Grid Heatmap:** The magnitudes of values shown through colors are laid out into a matrix of rows and columns, mostly by a density-based function. Below are the types of Grid Heatmaps.

- ***Clustered Heatmap:*** The goal of Clustered Heatmap is to build associations between both the data points and their features. This type of heatmap implements clustering as part of the process of grouping similar features. Clustered Heatmaps are widely used in biological sciences for studying gene similarities across individuals.

The order of the rows in Clustered Heatmap is determined by performing hierarchical cluster analysis of the rows. Clustering positions similar rows together on the map. Similarly, the order of the columns is determined.

- **Correlogram:** A correlogram replaces each of the variables on the two axes with numeric variables in the dataset. Each square depicts the relationship between the two intersecting variables, which helps to build descriptive or predictive statistical models.

- **Spatial Heatmap:** Each square in a Heatmap is assigned a color representation according to the nearby cells' value. The location of color is according to the magnitude of the value in that particular space. These Heatmaps are data-driven "paint by numbers" canvas overlaid on top of an image. The cells with higher values than other cells are given a hot color, while cells with lower values are assigned a cold color.

Correlation statistics:

Correlation

- Correlation measures the relationship between two variables.

We mentioned that a function has a purpose to predict a value, by converting input (x) to output (f(x)). We can say also say that a function uses the relationship between two variables for prediction.

Correlation Coefficient

The correlation coefficient measures the relationship between two variables.

The correlation coefficient can never be less than -1 or higher than 1.

- 1 = there is a perfect linear relationship between the variables (like Average_Pulse against Calorie_Burnage)

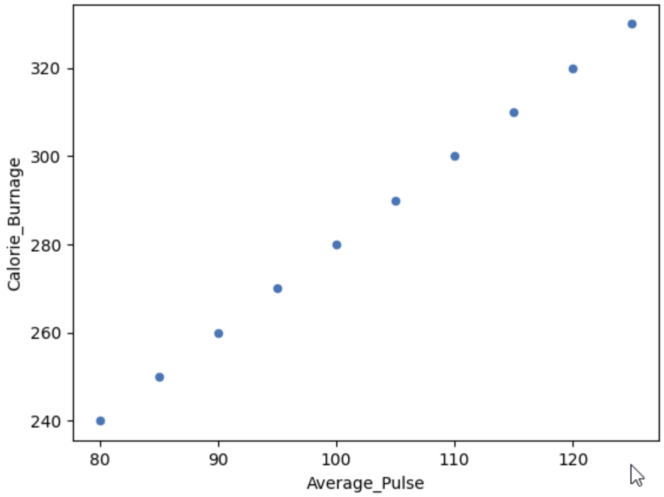
- 0 = there is no linear relationship between the variables
- -1 = there is a perfect negative linear relationship between the variables (e.g. Less hours worked, leads to higher calorie burnage during a training session)

Example of a Perfect Linear Relationship (Correlation Coefficient = 1)

- We will use scatterplot to visualize the relationship between Average_Pulse and Calorie_Burnage (we have used the small data set of the sports watch with 10 observations).
- This time we want scatter plots, so we change kind to "scatter":

```
#Three lines to make our compiler able to draw:
import sys
import matplotlib
matplotlib.use('Agg')

import pandas as pd
import matplotlib.pyplot as plt

health_data = pd.read_csv("data.csv", header=0, sep=",")

health_data.plot(x ='Average_Pulse', y='Calorie_Burnage', kind='scatter'),

plt.show()

#Two  lines to make our compiler able to draw:
plt.savefig(sys.stdout.buffer)
sys.stdout.flush()
```

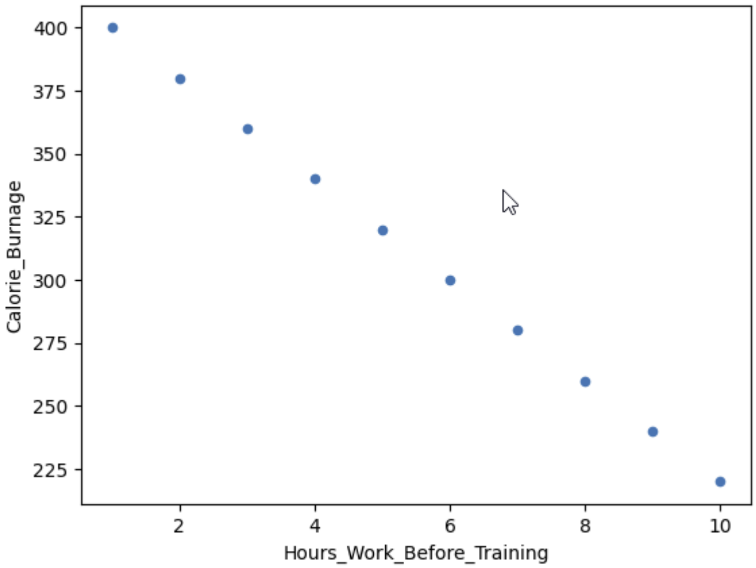
output:



Example of a Perfect Negative Linear Relationship (Correlation Coefficient = -1)

```
#Three lines to make our compiler able to draw:
import sys
import matplotlib
matplotlib.use('Agg')
import pandas as pd
import matplotlib.pyplot as plt
negative_corr = {'Hours_Work_Before_Training': [10,9,8,7,6,5,4,3,2,1],
'Calorie_Burnage': [220,240,260,280,300,320,340,360,380,400]}
negative_corr = pd.DataFrame(data=negative_corr)
negative_corr.plot(x ='Hours_Work_Before_Training', y='Calorie_Burnage', kind='scatter')
plt.show()
#Two  lines to make our compiler able to draw:
plt.savefig(sys.stdout.buffer)
sys.stdout.flush()
```

output:



We have plotted fictional data here. The x-axis represents the amount of hours worked at our job before a training session. The y-axis is Calorie_Burnage.

If we work longer hours, we tend to have lower calorie burnage because we are exhausted before the training session.

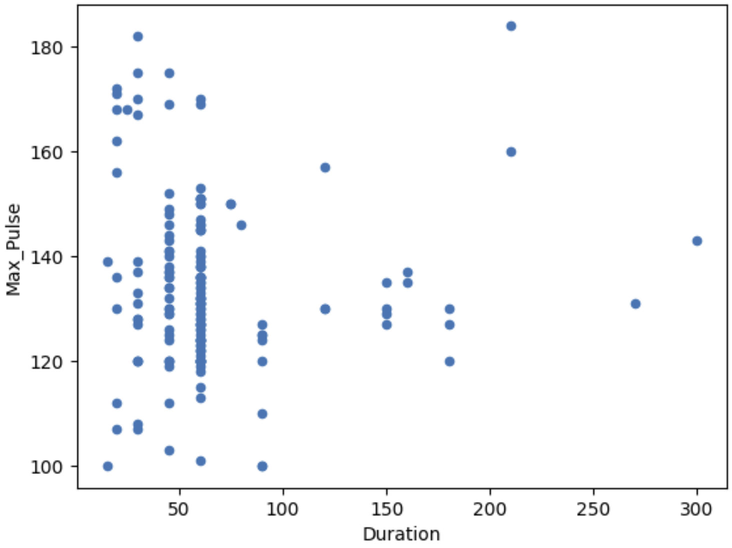
The correlation coefficient here is -1.

Example of No Linear Relationship (Correlation coefficient = 0)

```
#Three lines to make our compiler able to draw:

import sys

import matplotlib

matplotlib.use('Agg')

import pandas as pd

import matplotlib.pyplot as plt

full_health_data = pd.read_csv("data.csv", header=0, sep=",")
```

```
full_health_data.plot(x ='Duration', y='Max_Pulse', kind='scatter')

plt.show()

#Two lines to make our compiler able to draw:

plt.savefig(sys.stdout.buffer)

sys.stdout.flush()
```

output:



ANOVA:

ANOVA stands for analysis of variance and, as the name suggests, it helps us understand and compare variances among groups. Before going in detail about ANOVA, let’s remember a few terms in statistics:

- **Mean**: The average of all values.

- **Variance**: A measure of the variation among values. It is calculated by adding up squared differences of each value and the mean and then dividing the sum by the number of samples.

$$Variance = \frac{\sum(x_i - \text{mean})^2}{N}$$

X_i = value i

N = number of values

Mean = average of all values

- **Standard deviation**: The square root of variance.

In order to understand the motivation behind ANOVA, or some other statistical tests, we should learn two simple terms: population and sample.

Population is all elements in a group. For example,

- College students in US is a population that includes all of the college students in US.
- 25-year-old people in Europe is a population that includes all of the people that fits the description.

It is not always feasible or possible to do analysis on population because we cannot collect all the data of a population. Therefore, we use samples.

Sample is a subset of a population. For example,

- 1000 college students in US is a subset of “college students in US” population.

When we compare two samples (or groups), we can use **t-test** to see if there is any difference in means of groups. When we have more than two groups, t-test is not the optimal choice because we need to apply t-test to pairs separately. Consider we have groups A, B and C. To be able to compare the means, we need to apply a t-test to A-B, A-C and B-C. As the number of groups increase, this becomes harder to manage.

In the case of comparing three or more groups, ANOVA is preferred. There are two elements of ANOVA:

- Variation within each group
- Variation between groups

ANOVA test result is based on F ratio which is the ratio of the variation between groups to the variation within groups.

$$F = \frac{\text{Variation between groups}}{\text{Variation within groups}}$$

F ratio shows how much of the total variation comes from the variation between groups and how much comes from the variation within groups.

UNIT – 4

MODEL DEVELOPMENT

Regression Analysis

Regression analysis is a predictive modelling technique that assesses the relationship between dependent (i.e., the goal/target variable) and independent factors. Forecasting, time series modelling, determining the relationship between variables, and predicting continuous values can all be done using regression analysis. Just to give you an Analogy, Regression is the best way to study the relationship between household areas and a driver's household electricity cost.

Now, These Regression falls under 2 Categories Namely,

- **Simple Linear Regression:** The association between two variables is established using a straight line in Simple Linear Regression. It tries to create a line that is as near to the data as possible by determining the slope and intercept, which define the line and reduce regression errors. There is a single x and y variable

Equation: Y = mX+c

- **Multiple Linear Regression:** Multiple linear regressions are based on the presumption that both the dependent and independent variables, or Predictor and Target variables, have a linear relationship. There are two types of multilinear regressions: linear and nonlinear. It has one or more x variables and one or more y variables, or one dependent variable and two or more independent variables
    - Equation: Y = m1X1+m2X2+m3X3+..c
    - Where,
    - Y = Dependent Variable
      m = Slope
      X = Independent Variable
      c = Intercept
    - Now, let us understand both Simple and Multiple Linear Regression implementation with the below sample datasets!!

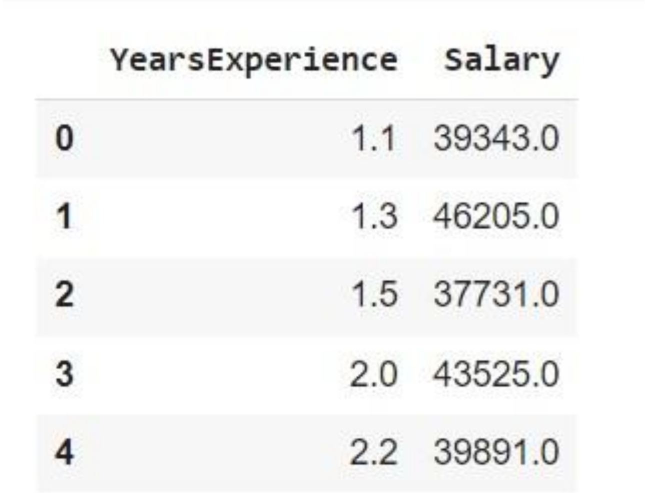
Simple Linear Regression

Given the experience let us predict the salary of the employees using a simple Regression model

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression
```

Read the data from the csv file

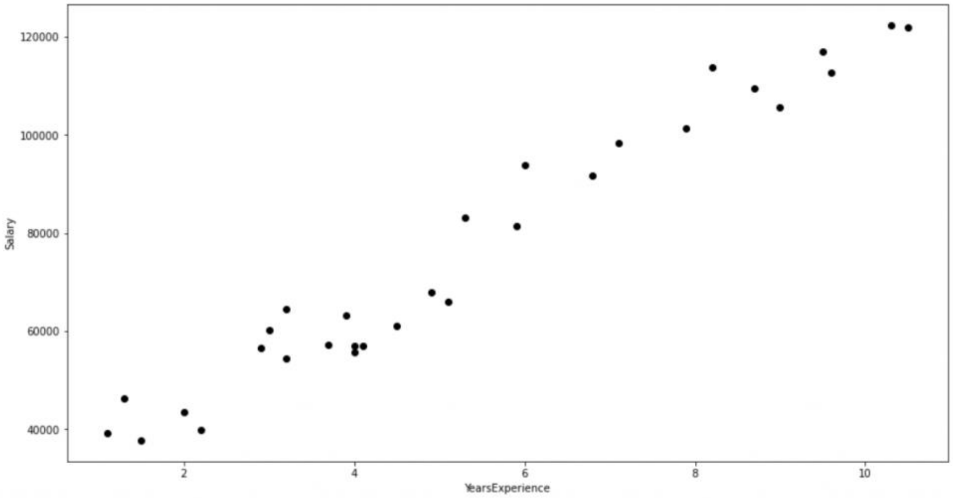
```
data = pd.read_csv('/content/Salary_Data.csv') #reading data
data.head()
```



	YearsExperience	Salary
0	1.1	39343.0
1	1.3	46205.0
2	1.5	37731.0
3	2.0	43525.0
4	2.2	39891.0

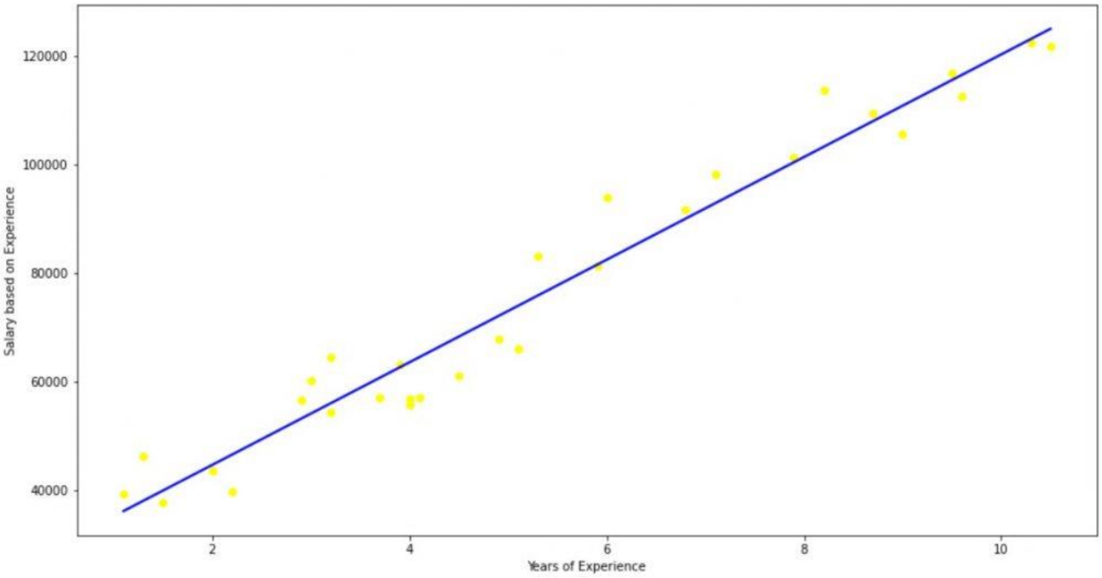
- **Plot Years of Experience vs Salary based on Experience**

```
plt.figure()
plt.scatter(data['YearsExperience'],data['Salary'],c='black')
plt.xlabel("YearsExperience")
plt.ylabel("Salary")
plt.show()
```



- **Reshape and fit the data into simple linear regression**

```
X = data['YearsExperience'].values.reshape(-1,1)
y = data['Salary'].values.reshape(-1,1)
reg = LinearRegression()
reg.fit(X, y)
Print R-squared value
```



Multiple Linear Regression

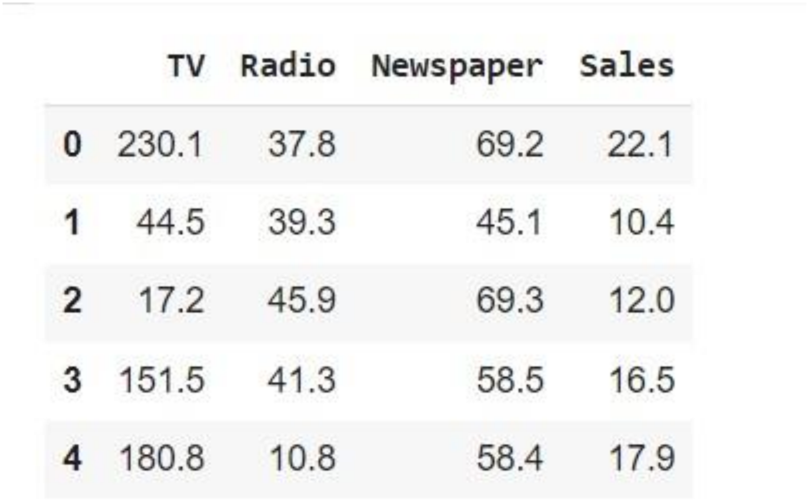
Let us the dataset set of advertising sales based on Tv and Newspaper

Import required libraries

```
%matplotlib inline
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
plt.rcParams['figure.figsize'] = (15.0, 8.0)
from mpl_toolkits.mplot3d import Axes3D
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error
from sklearn.metrics import r2_score
```

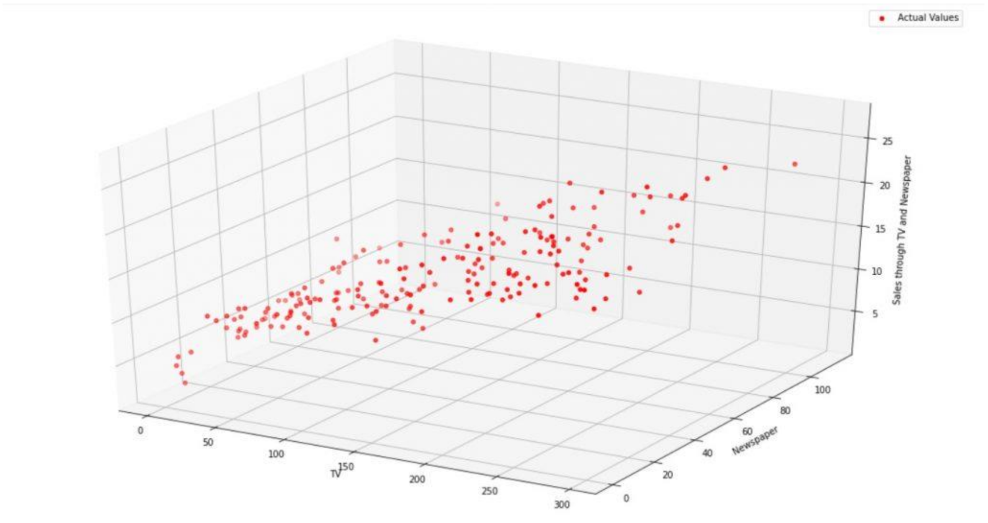
Read dataset from csv file

```
data = pd.read_csv('/content/Advertising Agency.csv') #reading data
data.head()
```



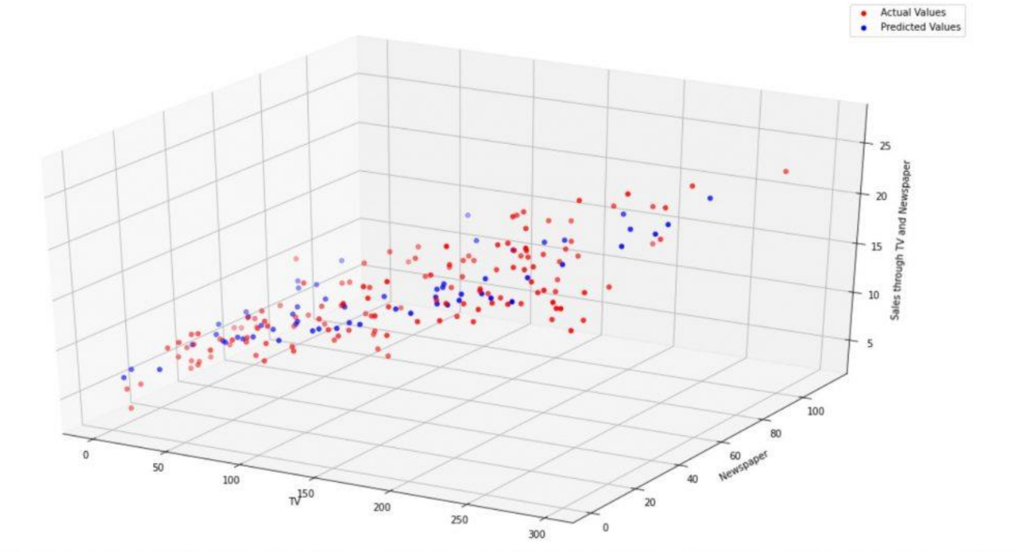
	TV	Radio	Newspaper	Sales
0	230.1	37.8	69.2	22.1
1	44.5	39.3	45.1	10.4
2	17.2	45.9	69.3	12.0
3	151.5	41.3	58.5	16.5
4	180.8	10.8	58.4	17.9

```
tv = data['TV'].values #storing dataframe values as variables
newspaper = data['Newspaper'].values
sales = data['Sales'].values
```



Plotting Actual vs Predicted values

```
fig = plt.figure(2)
ax = Axes3D(fig)
ax.scatter(X_train[:,0], X_train[:,1], y_train, color='r',label='Actual Values')
ax.scatter(X_test[:,0],X_test[:,1], y_pred, color='b',label='Predicted Values')
ax.set_xlabel('TV')
ax.set_ylabel('Newspaper')
ax.set_zlabel('Sales through TV and Newspaper')
ax.legend()
plt.show()
```



Multiple regression in machine learning:

Multiple Linear Regression is one of the important regression algorithms which models the linear relationship between a single dependent continuous variable and more than one independent variable.

Example:

Prediction of CO_2 emission based on engine size and number of cylinders in a car.

MLR equation:

In Multiple Linear Regression, the target variable(Y) is a linear combination of multiple predictor variables x_1, x_2, x_3, ...,x_n. Since it is an enhancement of Simple Linear Regression, so the same is applied for the multiple linear regression equation, the equation becomes:

Y= b<sub>0</sub>+b<sub>1</sub>x<sub>1</sub>+ b<sub>2</sub>x<sub>2</sub>+ b<sub>3 </sub>x<sub>3</sub>+...... bnxn

Where,

Y= Output/Response variable

b_0, b_1, b_2, b_3 , b_n....= Coefficients of the model.

x_1, x_2, x_3, x_4,...= Various Independent/feature variable

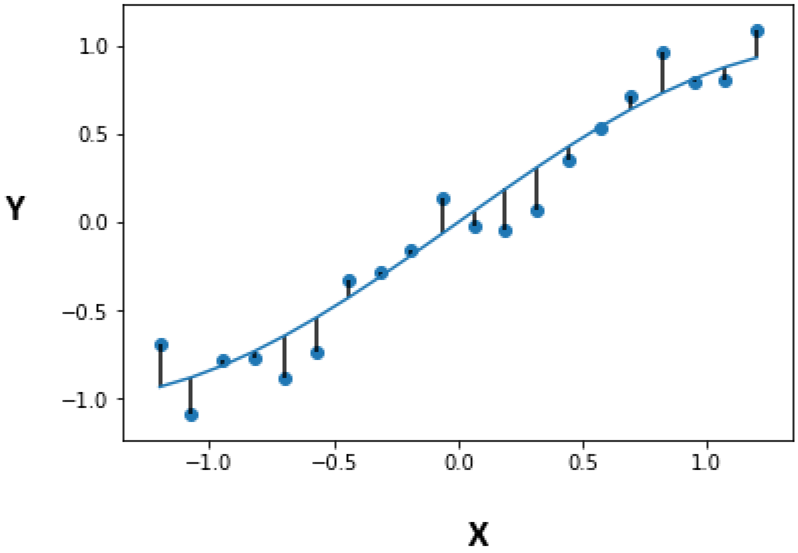
Model evaluation using visualization

Residual Plot:

Residuals

A residual is a measure of how far away a point is vertically from the regression line. Simply, it is the error between a predicted value and the observed actual value.

$$Residual\ (\epsilon) = y - \hat{y}$$

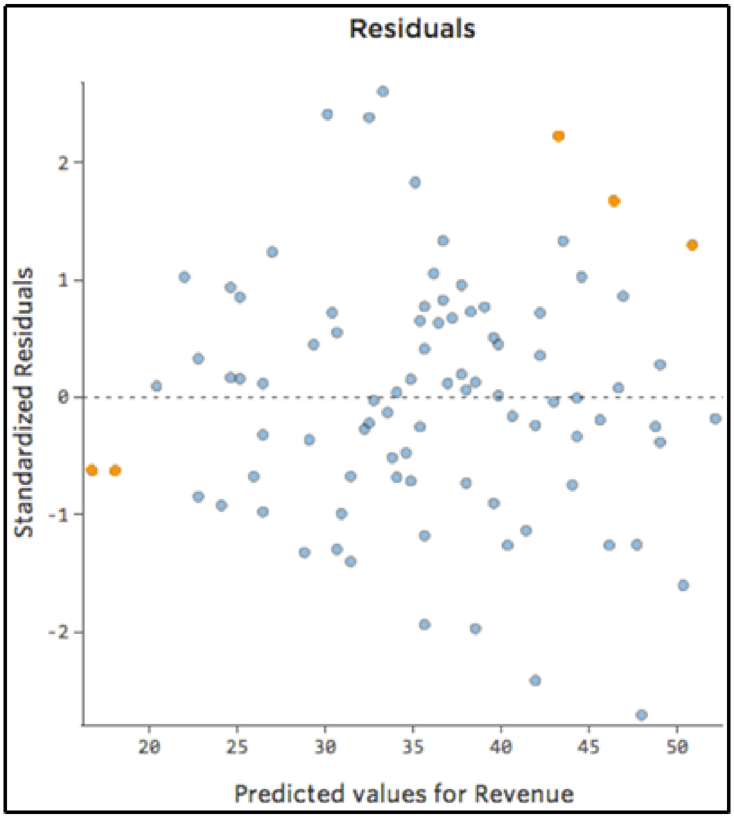


The above fig is an example of how to visualize residuals against the line of best fit. The vertical lines are the residuals.

Residual Plots

A typical residual plot has the residual values on the Y-axis and the independent variable on the x-axis. The below fig

is a good example of how a typical residual plot looks like.



Residual Plot Analysis

The most important assumption of a linear regression model is that the ***errors are independent and normally distributed.***

Let’s examine what this assumption means.

Every regression model inherently has some degree of error since you can never predict something 100% accurately. More importantly, randomness and unpredictability are always a part of the regression model. Hence, a regression model can be explained as:

$$\text{Response} = \text{Deterministic} + \text{Stochastic}$$

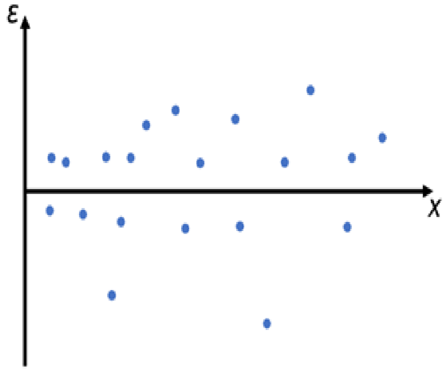
The deterministic part of the model is what we try to capture using the regression model. Ideally, our linear equation model should accurately capture the predictive information. Essentially, what this means is that if we capture all of the predictive information, all that is left behind (residuals) should be completely random & unpredictable i.e stochastic. Hence, we want our residuals to follow a normal distribution.

Characteristics of Good Residual Plots

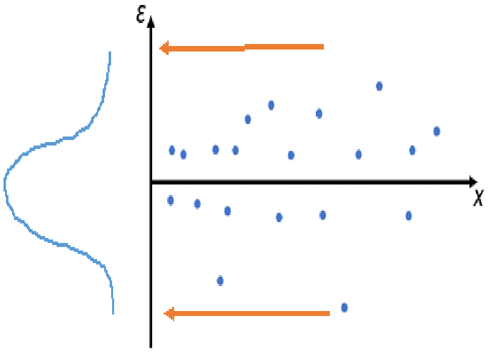
A few characteristics of a good residual plot are as follows:

1. It has a high density of points close to the origin and a low density of points away from the origin
2. It is symmetric about the origin

To explain why below fig is a good residual plot based on the characteristics above, we project all the residuals onto the y-axis. As seen in Figure 3b, we end up with a normally distributed curve; satisfying the assumption of the normality of the residuals.



Good residual plots

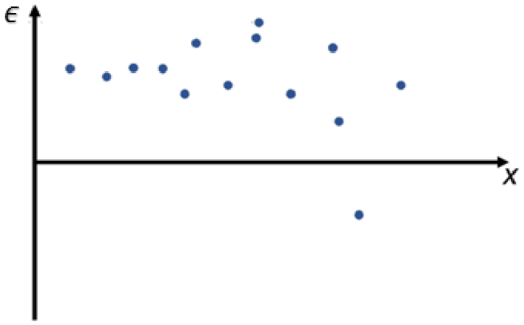


project on to the Y axis

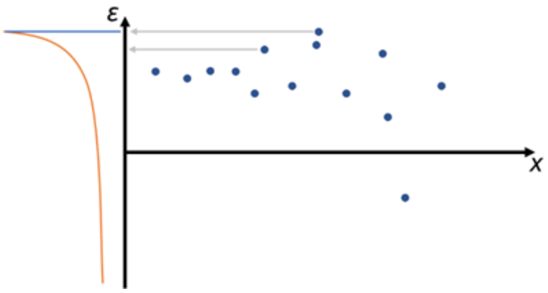
Finally, one other reason this is a good residual plot is, that independent of the value of an independent variable (x-axis), the residual errors are approximately distributed in the same manner. In other words, we **do not** see any patterns in the value of the residuals as we move along the x-axis.

Hence, this satisfies our earlier assumption that regression model **residuals are independent and normally distributed.**

Using the characteristics the bad residual plot: this plot has high density far away from the origin and low density close to the origin. Also, when we project the residuals on the y-axis, we can see the distribution curve is not normal.



Example of Bad Residual plot



Project onto the y-axis

Distribution plots

Polynomial Regression

In polynomial regression, the relationship between the independent variable x and the dependent variable y is described as an nth degree polynomial in x. Polynomial regression, abbreviated E(y |x), describes the fitting of a nonlinear relationship between the value of x and the conditional mean of y. It usually corresponded to the least-squares method.

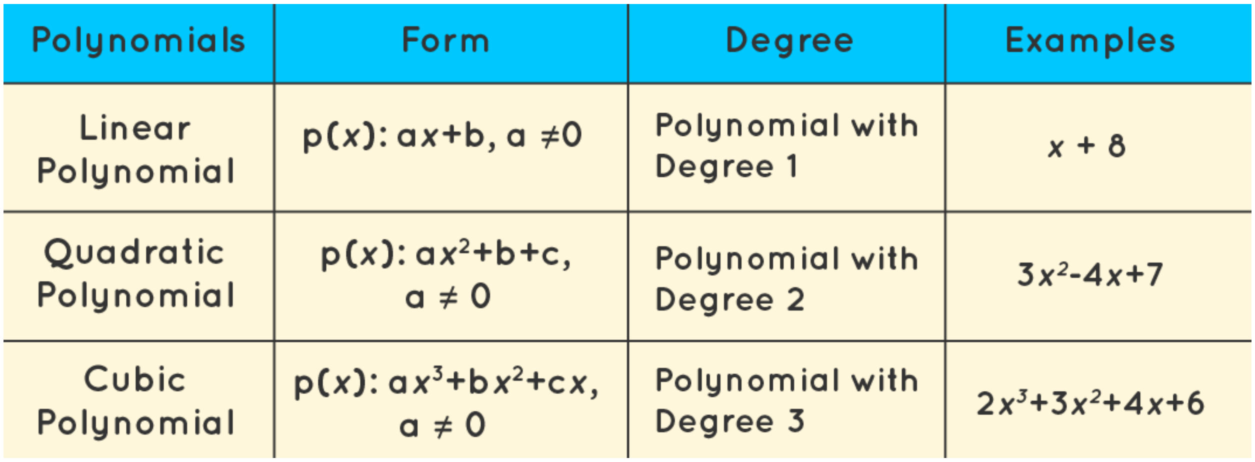
Types of Polynomial Regression

A quadratic equation is a general term for a second-degree polynomial equation. This degree, on the other hand, can go up to nth values. Polynomial regression can so be categorized as follows:

1. Linear – if degree as 1

2. Quadratic – if degree as 2

3. Cubic – if degree as 3 and goes on, on the basis of degree.



Polynomials	Form	Degree	Examples
Linear Polynomial	$p(x): ax+b, a \neq 0$	Polynomial with Degree 1	$x + 8$
Quadratic Polynomial	$p(x): ax^2+b+c, a \neq 0$	Polynomial with Degree 2	$3x^2-4x+7$
Cubic Polynomial	$p(x): ax^3+bx^2+cx, a \neq 0$	Polynomial with Degree 3	$2x^3+3x^2+4x+6$

Assumption of Polynomial Regression

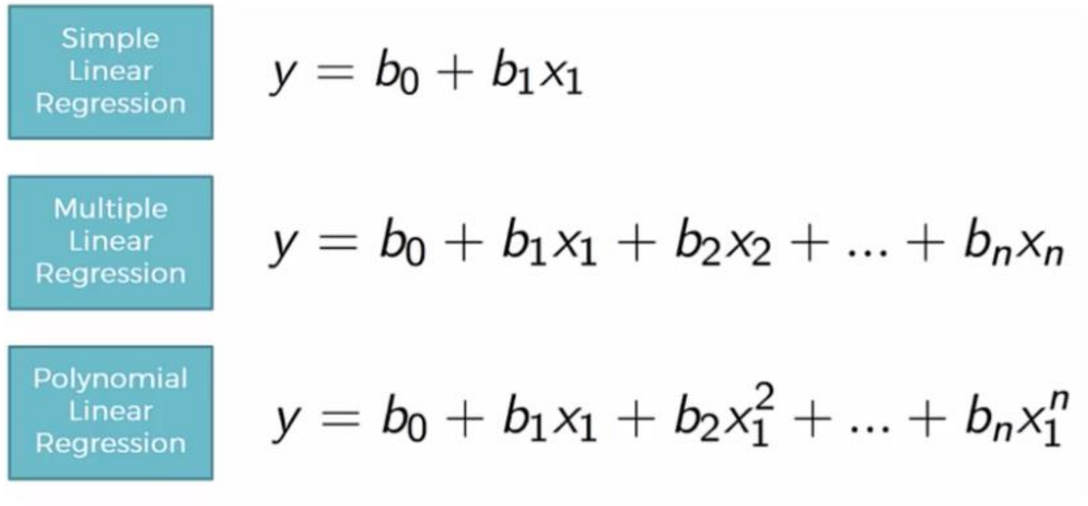
We cannot process all of the datasets and use polynomial regression machine learning to make a better judgment. We can still do it, but there should be specific constraints for the dataset in order to get the best polynomial regression results.

A dependent variable’s behaviour can be described by a linear, or curved, an additive link between the dependent variable and a set of k independent factors.

The independent variables have no relationship with one another.

We’re utilizing datasets with independent errors that are normally distributed with a mean of zero and a constant variance.

Here we are dealing with mathematics, rather than going deep, just understand the basic structure, we all know the equation of a linear equation will be a straight line, from that if we have many features then we opt for multiple regression just increasing features part alone, then how about polynomial, it’s not about increasing but changing the structure to a quadratic equation, you can visually understand from the diagram,



Simple Linear Regression	$y = b_0 + b_1x_1$
Multiple Linear Regression	$y = b_0 + b_1x_1 + b_2x_2 + \ldots + b_nx_n$
Polynomial Linear Regression	$y = b_0 + b_1x_1 + b_2x_1^2 + \ldots + b_nx_1^n$

Linear Regression Vs Polynomial Regression

Rather than focusing on the distinctions between linear and polynomial regression, we may comprehend the importance of polynomial regression by starting with linear regression. We build our model and realize that it performs abysmally. We examine the difference between the actual value and the best fit line we predicted, and it appears that the true value has a curve on the graph, but our line is nowhere near cutting the mean of the points. This is where polynomial regression comes into play; it predicts the best-fit line that matches the pattern of the data (curve).

One important distinction between Linear and Polynomial Regression is that Polynomial Regression does not require a linear relationship between the independent and dependent variables in the data set. When the Linear Regression Model fails to capture the points in the data and the Linear Regression fails to adequately represent the optimum conclusion, Polynomial Regression is used.

Non-linear data in Polynomial Regression

We need to enhance the model's complexity to overcome under-fitting. In this sense, we need to make linear analyzes in a non-linear way, statistically by using Polynomial,

$$y = \theta_0 + \theta_1 x \longrightarrow \theta_0 + \theta_1 x + \theta_2 x^2$$

Because the weights associated with the features are still linear, this is still called a linear model. x2 (x square) is only a function. However, the curve we're trying to fit is quadratic in nature.

Data science pipeline

A Data Science Pipeline is a collection of processes that transform raw data into actionable business answers. **Data Science Pipelines** automate the flow of data from source to destination, providing you with insights to help you make business decisions.

The Data Science Pipeline refers to the process and tools used to collect raw data from various sources, analyze it, and present the results in a Comprehensible Format. Companies use the process to answer specific business questions and generate actionable insights from real-world data. To find this information, all available Datasets, both External and Internal, are analyzed.

For example, your Sales Team would like to set realistic goals for the coming quarter. They can collect data from customer surveys or feedback, historical purchase orders, industry trends, and other sources using the data science pipeline. Robust data analysis tools are then used to thoroughly analyze the data and identify key trends and patterns.

Key Features of Data Science Pipelines

Here is a list of key features of the Data Science Pipeline:

- Continuous and Scalable Data Processing
- Cloud-based Elasticity and Agility.
- Data Processing Resources that are Self-Contained and Isolated.
- Access to a Large Amount of Data and the ability to self-serve.
- Disaster Recovery and High Availability

- Allow users to Delve into Insights at a Finer Level.
- Removes Data silos and Bottlenecks that cause Delays and Waste of Resources.

Working of Data Science Pipeline:

It is critical to have specific questions you want data to answer before moving raw data through the pipeline. This allows users to focus on the right data in order to uncover the right insights.

The Data Science Pipeline is divided into several stages, which are as follows:

- Obtaining Information
- Data Cleansing
- Data Exploration and Modeling
- Data Interpretation
- Data Revision

1) Obtaining Information

This is the location where data from internal, external, and third-party sources is collected and converted into a usable format (XML, JSON, .csv, etc.).

2) Data Cleansing

This is the most time-consuming step. Anomalies in data, such as duplicate parameters, missing values, or irrelevant information, must be cleaned before creating a data visualization.

3) Data Exploration and Modeling

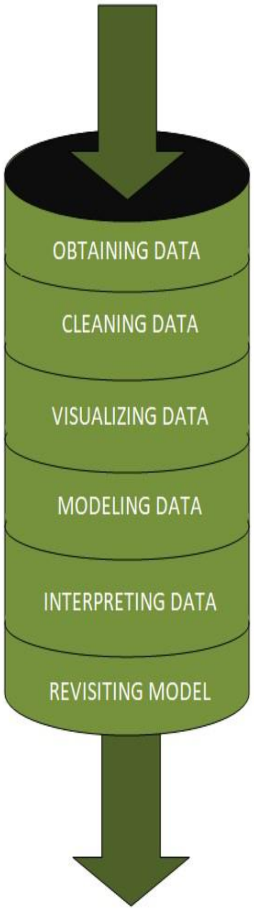
After thoroughly cleaning the data, it can be used to find patterns and values using data visualization tools and charts. This is where machine learning tools can help.

4) Data Interpretation

The goal of this step is to identify insights and then correlate them to your data findings. You can then use charts, dashboards, or reports to present your findings to business leaders or colleagues.

5) Data Revision

As business requirements change or more data becomes available, it’s critical to revisit your model and make any necessary changes.



Benefits:

Following are the benefits of Data Science Pipelines

1. **The pattern that can be replicated**
   Individual pipes are patterns in a larger architecture that may be recycled and reused for new data flows when data processing is viewed as a network of pipelines.
2. **Integration of new data sources takes less time.**
   Having a common concept and techniques for how data should pass through analytics systems makes it simpler to plan for integrating new data sources and minimizes the time and expense of integrating them.
3. **Data quality assurance**
   Understanding data streams as pipelines that need to be regulated and useful to end-users increases data quality and minimizes the chances of pipeline breakdowns going undiscovered.
4. **Assurance of the pipeline's security**
   With repetitive patterns and consistent knowledge of tools and architectures, security is baked in from the start. Good security procedures can easily apply to new dataflows or data sources.

5. **Build in stages**
   When you think of your dataflows as pipelines, you can scale them up gradually. You can get started early and achieve benefits immediately by starting with a modest controllable segment from a data source to a user.
6. **Agility and flexibility**
   Pipelines give a structure for responding dynamically to modifications in the sources or the needs of your data users.
   Extensible, modular, and reusable Data Pipelines are a bigger topic in Data Engineering that is very significant.

Features:

A well-designed end-to-end data science pipeline can find, collect, manage, analyze, model, and transform data to uncover possibilities and create cost-effective business operations.

Current data science pipelines make extracting knowledge from the big data you collect simple and quick.

The finest data science pipelines contain the following features to accomplish this:

- Data processing that is both continuous and expandable
- Elasticity and agility afforded by the cloud
- Access to data on a large scale and the capacity to self-serve
- Disaster recovery and high availability

MEASURES FOR IN – SAMPLE EVALUATION:

Measures for in – sample evaluation :

A way to numerically determine how good the model fits the dataset.

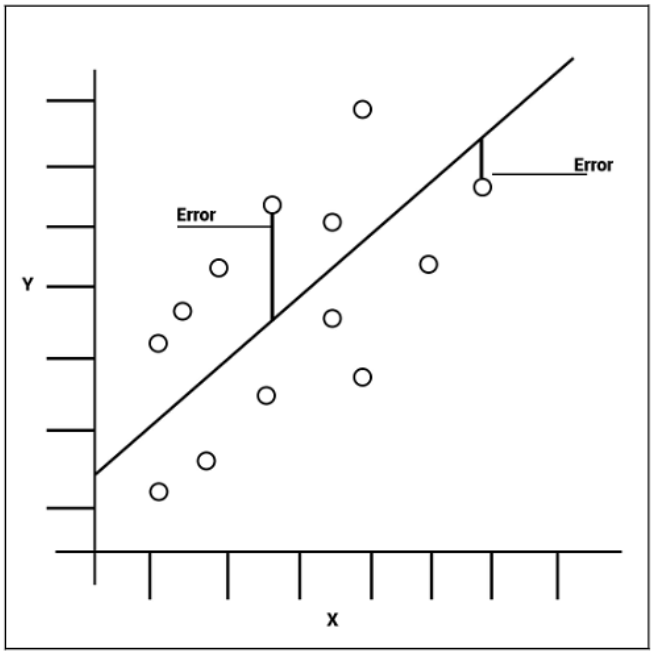
Two important measures to determine the fit of a model:

- Mean squared error(MSE)
- R squared ($R^{\wedge 2}$)

Mean squared error(MSE)

The Mean Squared Error measures how close a regression line is to a set of data points. It is a risk function corresponding to the expected value of the squared error loss.

Mean square error is calculated by taking the average, specifically the mean, of errors squared from data as it relates to a function.



A larger MSE indicates that the data points are dispersed widely around its central moment (mean), whereas a smaller MSE suggests the opposite. A smaller MSE is preferred because it indicates that your data points are dispersed closely around its central moment (mean). It reflects the centralized distribution of your data values, the fact that it is not skewed, and, most importantly, it has fewer errors (errors measured by the dispersion of the data points from its mean).

Lesser the MSE => Smaller is the error => Better the estimator.

The Mean Squared Error is calculated as:

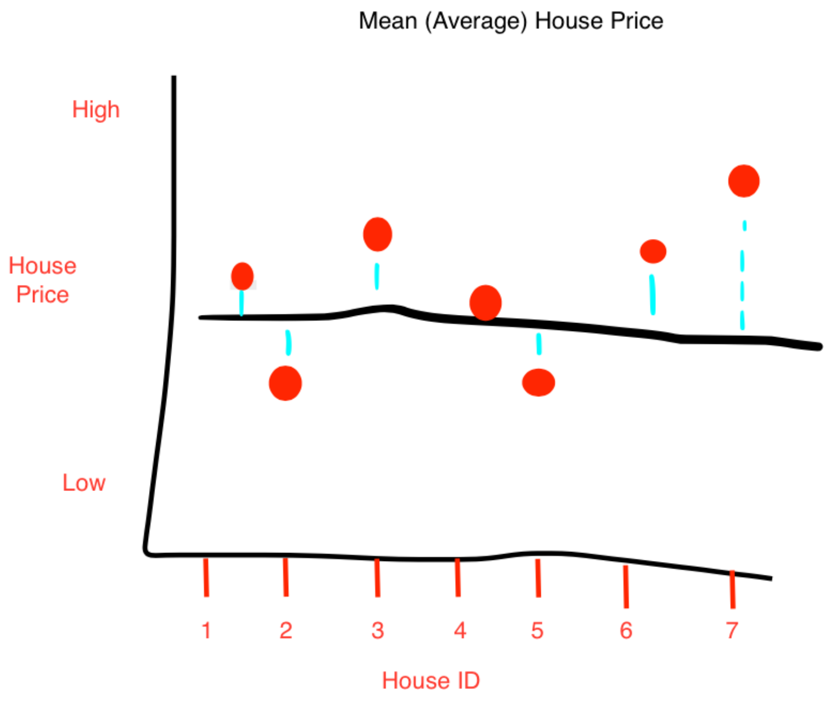
MSE = (1/n) * Σ(actual – forecast)2

where:

- Σ – a symbol that means “sum”
- n – sample size
- actual – the actual data value
- forecast – the predicted data value

R squared ($R^{\wedge 2}$)

R-squared is a metric of correlation. Correlation is measured by **“r”** and it tells us how strongly two variables can be related. A correlation closer to +1 means a strong relationship in the positive direction, while -1 means a stronger relationship in the opposite direction. A value closer to 0 means that there is not much of a relationship between the variables. R-squared is closely related to correlation.



The best way to understand **R-squared** is through a simple example. In this example, the black horizontal line represents the **mean** price of houses. The vertical blue line represents the **variation.** Variation means the difference between each point and the mean. The variation of the data can be calculated by the sum of the squared difference for each point minus the mean.

In other words: **Variation = Sum(HousePrice *i*-Mean)²**

UNIT V

MODEL EVALUATION

GENERALIZATION ERROR

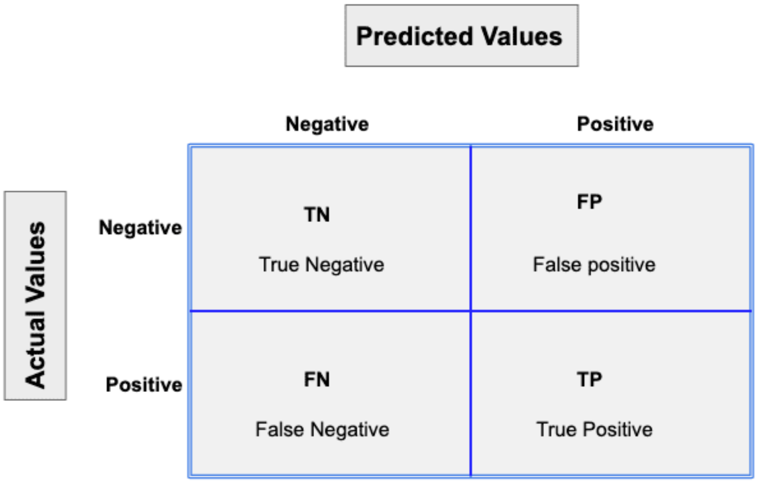
EVALUATION METRICS

Model Evaluation Metrics define the evaluation metrics for evaluating the performance of a machine learning model, which is an integral component of any data science project. It aims to estimate the generalization accuracy of a model on the future (unseen/out-of-sample) data.

Confusion Matrix

A confusion matrix is a matrix representation of the prediction results of any binary testing that is often used to **describe the performance of the classification model** (or “classifier”) on a set of test data for which the true values are known.

The confusion matrix itself is relatively simple to understand, but the related terminology can be confusing.



		Predicted Values	
		Negative	Positive
Actual Values	Negative	TN True Negative	FP False positive
	Positive	FN False Negative	TP True Positive

Each prediction can be one of the four outcomes, based on how it matches up to the actual value:

- True Positive (TP): Predicted True and True in reality.
- True Negative (TN): Predicted False and False in reality.
- False Positive (FP): Predicted True and False in reality.
- False Negative (FN): Predicted False and True in reality.

Now let us understand this concept using hypothesis testing.

A Hypothesis is speculation or theory based on insufficient evidence that lends itself to further testing and experimentation. With further testing, a hypothesis can usually be proven true or false.

A Null Hypothesis is a hypothesis that says there is no statistical significance between the two variables in the hypothesis. It is the hypothesis that the researcher is trying to disprove.

We would always reject the null hypothesis when it is false, and we would accept the null hypothesis when it is indeed true.

Even though hypothesis tests are meant to be reliable, there are two types of errors that can occur.

These errors are known as Type 1 and Type II errors.

For example, when examining the effectiveness of a drug, the null hypothesis would be that the drug does not affect a disease.

Type I Error:- equivalent to False Positives(FP).

The first kind of error that is possible involves the rejection of a null hypothesis that is true.

Let’s go back to the example of a drug being used to treat a disease. If we reject the null hypothesis in this situation, then we claim that the drug does have some effect on a disease. But if the null hypothesis is true, then, in reality, the drug does not combat the disease at all. The drug is falsely claimed to have a positive effect on a disease.

Type II Error:- equivalent to False Negatives(FN).

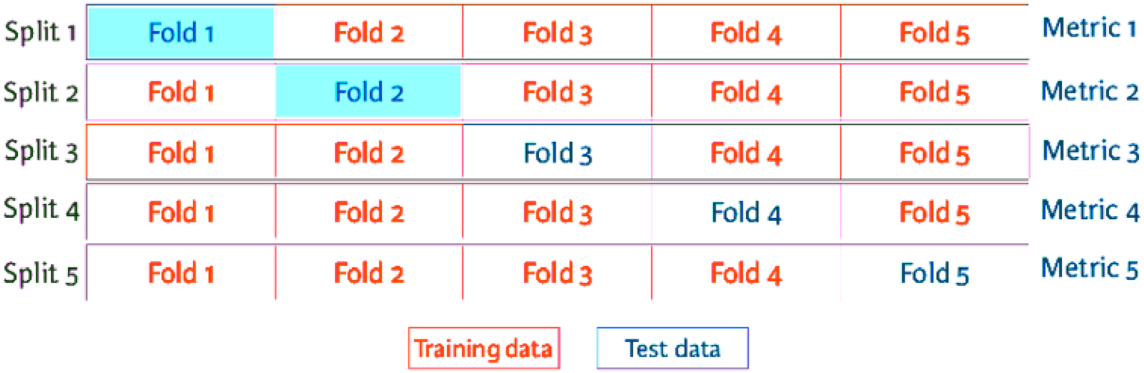
The other kind of error that occurs when we accept a false null hypothesis. This sort of error is called a type II error and is also referred to as an error of the second kind.

CROSS VALIDATION

Cross validation is a technique for assessing how the statistical analysis generalises to an independent data set.It is a technique for evaluating machine learning models by training several models on subsets of the available input data and evaluating them on the complementary subset of the data. Using cross-validation, there are high chances that we can detect over-fitting with ease.

K-Fold Cross Validation

First I would like to introduce you to a golden rule — *“Never mix training and test data”*. Your first step should always be to **isolate the test data-set** and use it only for final evaluation. Cross-validation will thus be performed on the training set.


Split 1 Fold 1 Fold 2 Fold 3 Fold 4 Fold 5 Metric 1
Split 2 Fold 1 Fold 2 Fold 3 Fold 4 Fold 5 Metric 2
Split 3 Fold 1 Fold 2 Fold 3 Fold 4 Fold 5 Metric 3
Split 4 Fold 1 Fold 2 Fold 3 Fold 4 Fold 5 Metric 4
Split 5 Fold 1 Fold 2 Fold 3 Fold 4 Fold 5 Metric 5
Training data
Test data

Initially, the entire training data set is broken up in *k* equal parts. The first part is kept as the hold out (testing) set and the remaining *k-1* parts are used to train the model. Then the trained model is then tested on the holdout set. The above process is repeated k times, in each case we keep on changing the holdout set. Thus, every data point get an equal opportunity to be included in the test set.

Usually, k is equal to 3 or 5. It can be extended even to higher values like 10 or 15 but it becomes extremely computationally expensive and time-consuming. Let us have a look at how we can implement this with a few lines of Python code and the Sci-kit Learn API.

```
from sklearn.model_selection import cross_val_score
print(cross_val_score(model, X_train, y_train, cv=5))
```

We pass the **model** or classifier object, the features, the labels and the parameter **cv** which indicates the **K** for K-Fold cross-validation. The method will return a list of k accuracy values for each iteration. In general, we take the average of them and use it as a consolidated cross-validation score.

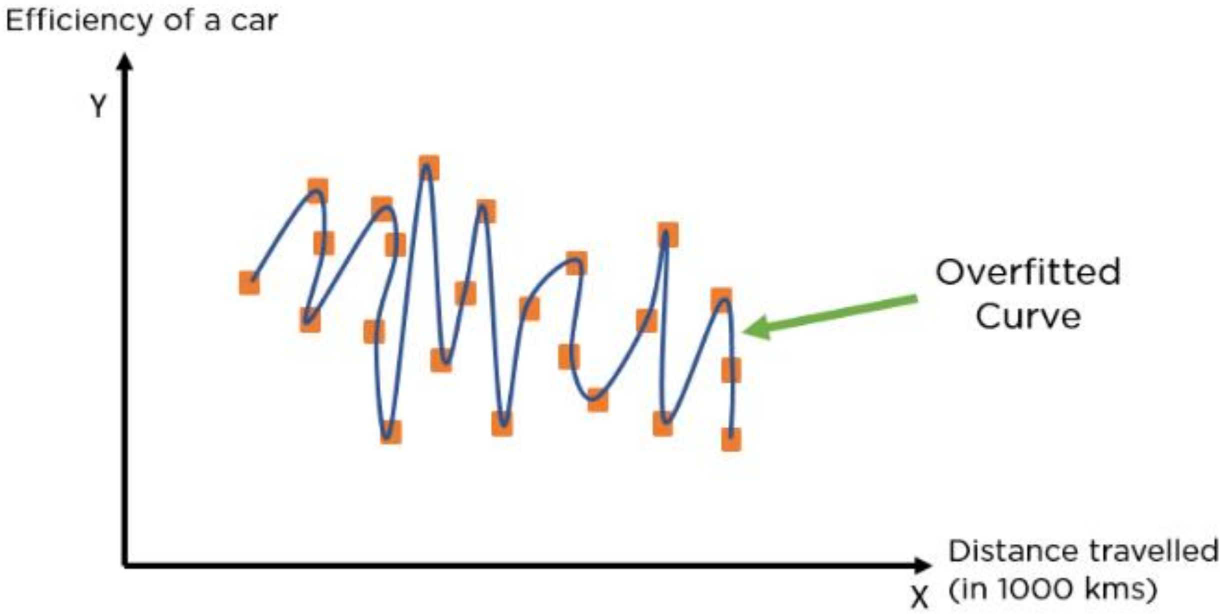
```
        import numpy as np
print(np.mean(cross_val_score(model, X_train, y_train, cv=5)))
```

Although it might be computationally expensive, cross-validation is essential for evaluating the performance of the learning model.

Overfitting and Underfitting :

What is Overfitting?

When a model performs very well for training data but has poor performance with test data (new data), it is known as overfitting. In this case, the machine learning model learns the details and noise in the training data such that it negatively affects the performance of the model on test data. Overfitting can happen due to low bias and high variance.



Reasons for Overfitting

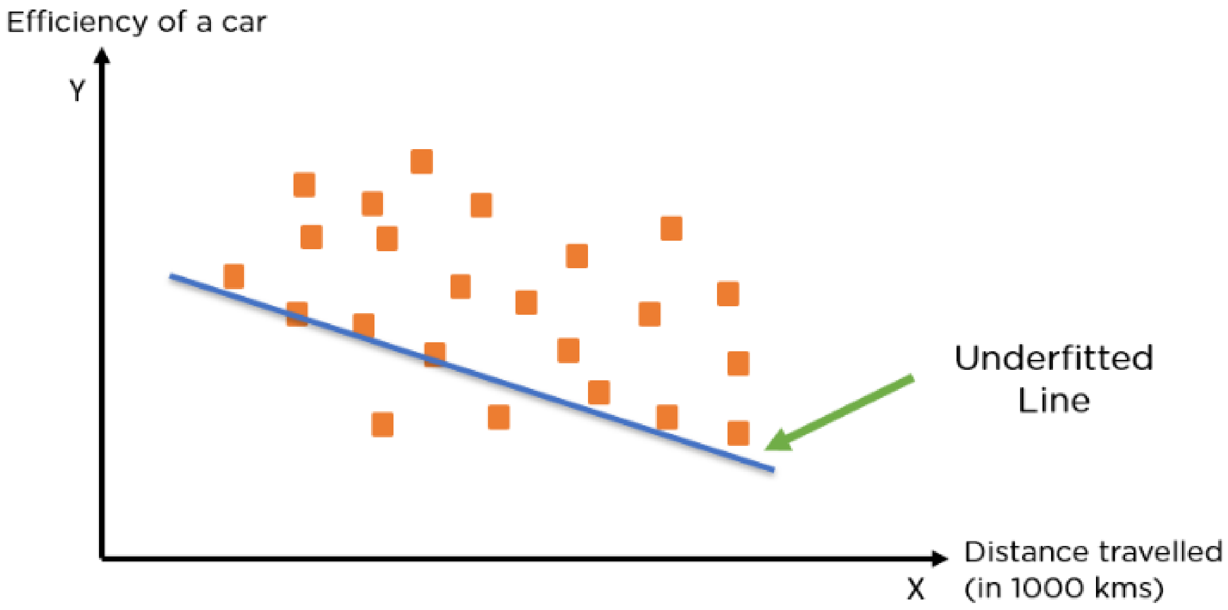
- Data used for training is not cleaned and contains noise (garbage values) in it
- The model has a high variance
- The size of the training dataset used is not enough
- The model is too complex

Ways to Tackle Overfitting

- Using K-fold cross-validation
- Using Regularization techniques such as Lasso and Ridge
- Training model with sufficient data
- Adopting ensembling techniques

What is Underfitting?

When a model has not learned the patterns in the training data well and is unable to generalize well on the new data, it is known as underfitting. An underfit model has poor performance on the training data and will result in unreliable predictions. Underfitting occurs due to high bias and low variance.



Reasons for Underfitting

- Data used for training is not cleaned and contains noise (garbage values) in it
- The model has a high bias
- The size of the training dataset used is not enough
- The model is too simple

Ways to Tackle Underfitting

- Increase the number of features in the dataset
- Increase model complexity
- Reduce noise in the data
- Increase the duration of training the data

Ridge Regression

Ridge regression is a model tuning method that is used to analyse any data that suffers from multicollinearity. This method performs L2 regularization. When the issue of multicollinearity occurs, least-squares are unbiased, and variances are large, this results in predicted values being far away from the actual values.

The cost function for ridge regression:

Min(||Y – X(theta)||^2 + λ||theta||^2)

Lambda is the penalty term. λ given here is denoted by an alpha parameter in the ridge function.

Ridge Regression Models

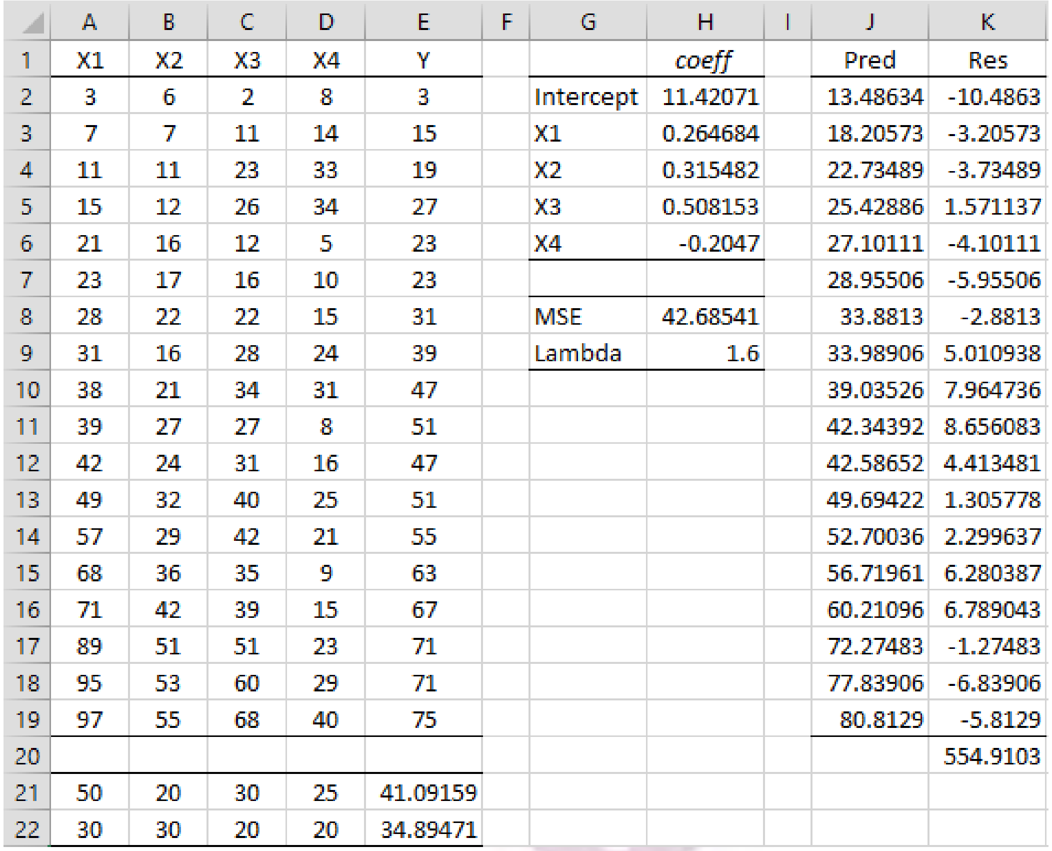
For any type of regression machine learning model, the usual regression equation forms the base which is written as:

Y = XB + e

Where Y is the dependent variable, X represents the independent variables, B is the regression coefficients to be estimated, and e represents the errors are residuals.

Ridge Regression Predictions

We now show how to make predictions from a Ridge regression model. In particular, we will make predictions based on the Ridge regression model created for Example 1 with lambda = 1.6. The raw input data is repeated in range A1:E19 of Figure 1 and the unstandardized regression coefficients calculated in Figure 2 of Ridge Regression Analysis Tool is repeated in range G2:H6 of Figure 1.



	A	B	C	D	E	F	G	H	I	J	K
1	X1	X2	X3	X4	Y			*coeff*		Pred	Res
2	3	6	2	8	3		Intercept	11.42071		13.48634	-10.4863
3	7	7	11	14	15		X1	0.264684		18.20573	-3.20573
4	11	11	23	33	19		X2	0.315482		22.73489	-3.73489
5	15	12	26	34	27		X3	0.508153		25.42886	1.571137
6	21	16	12	5	23		X4	-0.2047		27.10111	-4.10111
7	23	17	16	10	23					28.95506	-5.95506
8	28	22	22	15	31		MSE	42.68541		33.8813	-2.8813
9	31	16	28	24	39		Lambda	1.6		33.98906	5.010938
10	38	21	34	31	47					39.03526	7.964736
11	39	27	27	8	51					42.34392	8.656083
12	42	24	31	16	47					42.58652	4.413481
13	49	32	40	25	51					49.69422	1.305778
14	57	29	42	21	55					52.70036	2.299637
15	68	36	35	9	63					56.71961	6.280387
16	71	42	39	15	67					60.21096	6.789043
17	89	51	51	23	71					72.27483	-1.27483
18	95	53	60	29	71					77.83906	-6.83906
19	97	55	68	40	75					80.8129	-5.8129
20											554.9103
21	50	20	30	25	41.09159						
22	30	30	20	20	34.89471						

The predictions for the input data are shown in column J. In fact, the values in range J2:J19 can be calculated by the array formula

=H2+MMULT(A2:D19,H3:H6).

Alternatively, they can be calculated by the array formula

=RidgePred(A2:D19,A2:D19,E2:E19,H9)

Real Statistics Function: The Real Statistics Resource Pack provides the following functions.

RidgeMSE(Rx, Ry, *lambda*) = MSE of the Ridge regression defined by the *x* data in Rx, y data in Ry and the given lambda value.

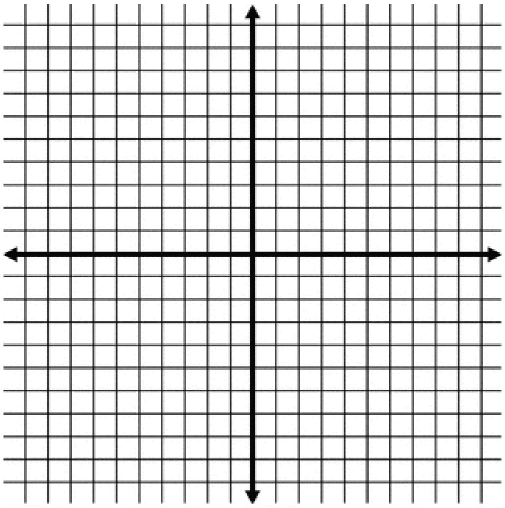
RidgePred(Rx0, Rx, Ry, *lambda*): returns an array of predicted y values for the *x* data in range Rx0 based on the Ridge regression model defined by Rx, Ry and *lambda*; if Rx0 contains only one row then only one y value is returned.

GRID SEARCH

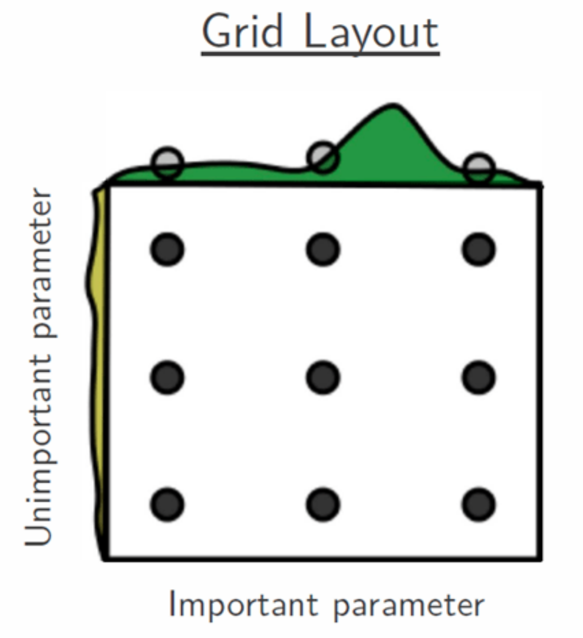
Grid-search is used to find the optimal *hyperparameters* of a model which results in the most 'accurate' predictions.

Grid search refers to a technique used to identify the optimal hyperparameters for a model. Unlike parameters, finding hyperparameters in training data is unattainable. As such, to find the right hyperparameters, we create a model for each combination of hyperparameters.

Grid search is thus considered a very traditional hyperparameter optimization method since we are basically "brute-forcing" all possible combinations. The models are then evaluated through cross-validation. The model boasting the best accuracy is naturally considered to be the best.



A model ***hyperparameter*** is a characteristic of a model that is external to the model and whose value cannot be estimated from data. The value of the hyperparameter has to be set before the learning process begins. For example, *c* in Support Vector Machines, *k* in k-Nearest Neighbors, *the number of hidden layers* in Neural Networks.



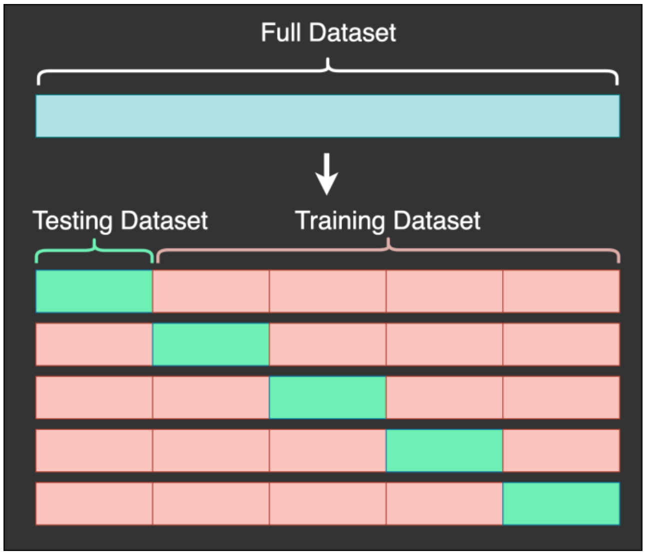
Cross validation

We have mentioned that cross-validation is used to evaluate the performance of the models. Cross-validation measures how a model generalizes itself to an independent dataset. We use cross-validation to get a good estimate of how well a predictive model performs.

With this method, we have a pair of datasets: an independent dataset and a training dataset. We can partition a single dataset to yield the two sets. These partitions are of the same size and are referred to as folds. A model in consideration is trained on all folds, bar one.

The excluded fold is used to then test the model. This process is repeated until all folds are used as the test set. The average performance of the model on all folds is then used to estimate the model’s performance.

In a technique known as the k-fold cross-validation, a user specifies the number of folds, represented by kk. This means that when k=5k=5, there are 5 folds.



K-fold cross-validation with K as 5.

Grid search implementation

The example given below is a basic implementation of grid search. We first specify the hyperparameters we seek to examine. Then we provide a set of values to test.

1. **Load dataset**.

My first step is loading the dataset using from sklearn.datasets import load_iris and iris = load_iris(). The iris dataset is sci-kit learn library in Python. Data is stored in a 150∗4150∗4 array.

2. **Import GridSearchCV, svm and SVR.**

After loading the dataset, we then import GridSearchCV as well as svm and SVR from sklearn.model_selection

```
sklearn.model_selection import GridSearchCV

from sklearn import svm

from sklearn.svm import SVR
```

3. **Set estimator parameters.**

In this implementation, we use the rbf kernel of the SVR model. rbf stands for the radial basis function. It introduces some form of non-linearity to the model since the data in use is non-linear. By this, we mean that the data arrangement follows no specific sequence.

```
estimator=SVR(kernel='rbf')
```

4. **Specify hyperparameters and range of values.**

We then specify the hyperparameters we seek to examine. When using the SVR’s rbf kernel, the three hyperparameters to use are C, epsilon, and gamma. We can give each one several values to choose from.

5. **Evaluation.**

We mentioned that cross-validation is carried out to estimate the performance of a model. In k-fold cross-validation, k is the number of folds. As shown below, through cv=5, we use cross-validation to train the model 5 times. This means that 5 would be the kk value.

6. **Fitting the data.**

We do this through grid.fit(X,y), which does the fitting with all the parameters.