SUMMER TRAINING ON BASIC OF DATA SCIENCE

BY UPGRAD

A TRAINING REPORT

Submitted in partial fulfilment of the requirements for the award of degree of

B.TECH. COMPUTER SCIENCE (DATA SCIENCE)

SUBMITTED TO

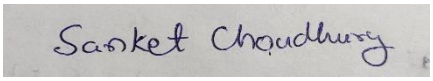
LOVELY PROFESSIONAL UNIVERSITY, PHAGWARA, PUNJAB

FROM 05/24/2021 TO 07/15/2021

SUBMITTED BY

Name of student: SANKET CHOUDHURY

Registration No.: 11902359

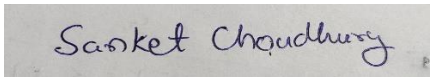
Signature of student: 

DECLARATION

To whom so ever it may concern

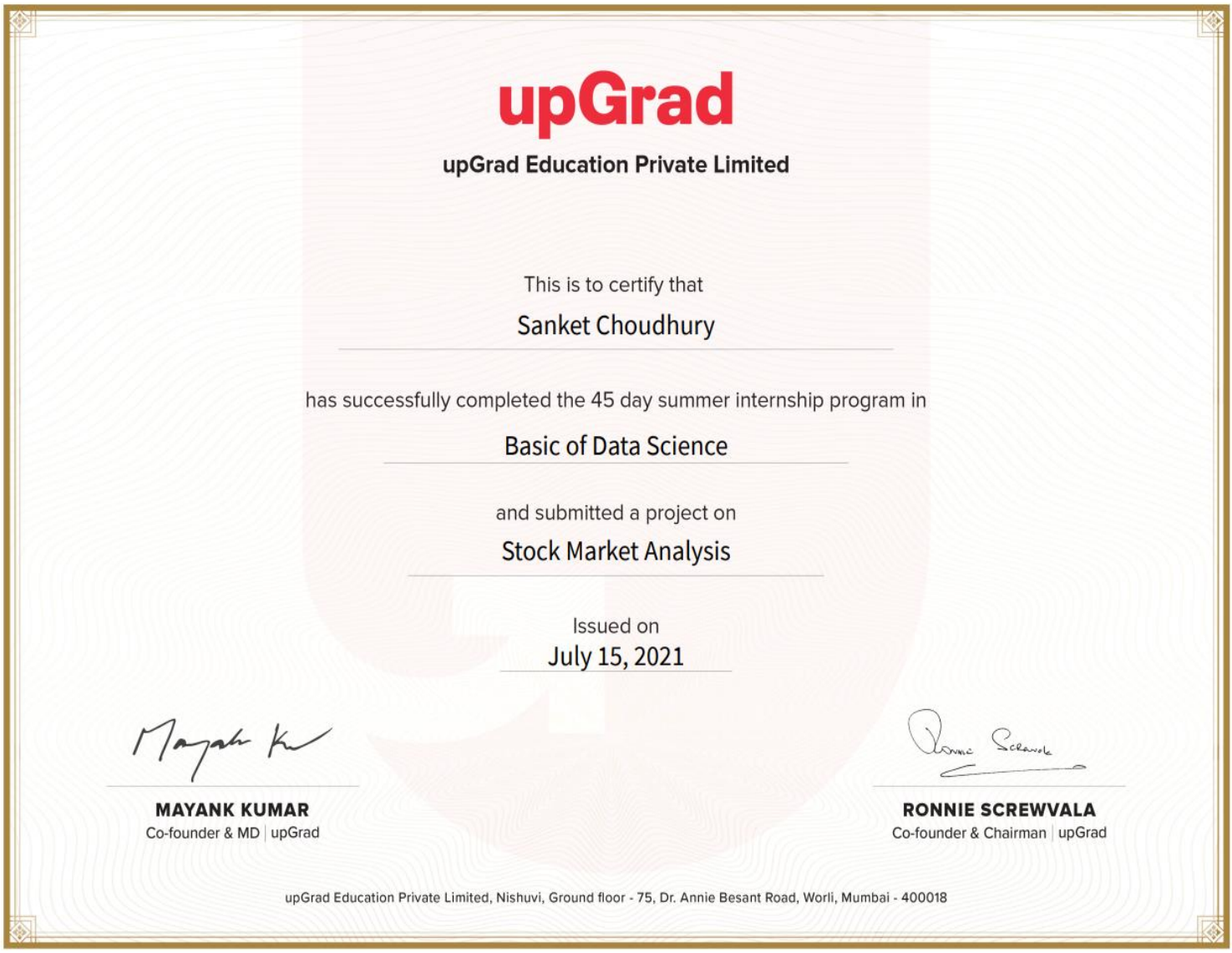
I**, Sanket Choudhury**, **11902359**, hereby declare that the work done by me on the course of "**Basic of Data Science by UpGrad**" from **May 24th, 2021,** to **July 15th, 2021**, is a record of original work for the partial fulfillment of the requirements for the award of the degree, "**Bachelor of Technology (B.Tech.) in CSE"**.

Sanket Choudhury (11902359)



Date - 3rd September 2021

SUMMER TRAINING CERTIFICATE



upGrad

upGrad Education Private Limited

This is to certify that

Sanket Choudhury

has successfully completed the 45 day summer internship program in

Basic of Data Science

and submitted a project on

Stock Market Analysis

Issued on

July 15, 2021

MAYANK KUMAR
Co-founder & MD | upGrad

RONNIE SCREWVALA
Co-founder & Chairman | upGrad

upGrad Education Private Limited, Nishuvi, Ground floor - 75, Dr. Annie Besant Road, Worli, Mumbai - 400018

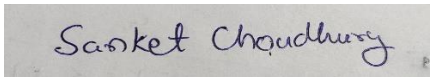
ACKNOWLEDGEMENT

The success and final outcome of learning Data Science required a lot of guidance and assistance from many people and I am extremely privileged to have got this all along the completion of my course and few of the projects. All that I have done is only due to such supervision and assistance and I would not forget to thank them.

I respect and thank **UpGrad**, for providing me an opportunity to do the course and project work and giving me all support and guidance, which made me complete the course duly. I am extremely thankful to the course advisor **Mr. Santosh Kumar** and **Mr. Ashish Saha**.

I am thankful to and fortunate enough to get constant encouragement, support, and guidance from all Teaching staffs of UpGrad which helped me in successfully completing my course and project work.

Sanket Choudhury (11902359)



3rd September 2021

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES/CHARTS

LIST OF ABBREVIATIONS

1. MA (Moving average)
2. SQL (Structured Query Language)
3. RDBMS (Relational Database Management System)

CHAPTER I
INTRODUCTION OF THE PROJECT UNDERTAKEN

The project undertaken during this 8-week summer training on "BASIC OF DATA SCIENCE" was on "**STOCK MARKET ANALYSIS".**

In this project, we were provided with stock price data from 1-Jan-2015 to 31-July-2018 for six stocks Eicher Motors, Hero, Bajaj Auto, TVS Motors, Infosys and TCS.

With the help of some knowledge of "Moving Average", and these data sets we had to complete 5 tasks as provided by the company and it was solely done on MySQL platform.

The 5 tasks were:

Task 1: Create a new table named 'bajaj1' containing the date, close price, 20 Day MA and 50 Day MA. (This must be done for all 6 stocks)

Date	Close Price	20 Day MA	50 Day MA

Task 2: Create a master table containing the date and close price of all the six stocks. (Column header for the price is the name of the stock)

The table header should look like:

Date	Bajaj	TCS	TVS	Infosys	Eicher	Hero

Task 3: Use the table created in Task(1) to generate buy and sell signal. Store this in another table named 'bajaj2'. Perform this operation for all stocks.

Date	Close Price	Signal

Task 4: Create a User defined function, that takes the date as input and returns the signal for that particular day (Buy/Sell/Hold) for the Bajaj stock. (**Hint**: The signal of sell and buy for that particular day is generated by subtracting the previous day's flag value. Flag value is generated using short-term and long-term moving averages.)

Task 5: Make an analysis out of the the whole to get the trend of the stocks and write a brief summary of the results obtained and what inferences you can draw from the analysis performed.

So , these were the task to be completed for the project.

So , now the big question comes how is this important and how does it help someone?

This question will be answered after I just briefly describe the word , “Moving average”.

MOVING AVERAGE :

Two of India's biggest stock exchanges BSE and NSE, collectively clear trades combining to greater than 40,000 crores every day. As you might already be aware, a lot of trading happens on the basis of technical and fundamental analysis.

One of the most basic technical analysis used by a lot of stock traders is the **Moving Average Method**.

Consider the following price trend of a particular stock.

Week 1: 13,14,11,17,19

Week 2: 26,23,22,22,14

Week 3: 17,19,13,16,17

Day	Price	10 Day	Value used for 10 Day MA	5 Day	Value used for 5 Day MA
1	13				
2	14				
3	11				
4	17				
5	19			14.8	Average of Day 1 Through 5
6	26			17.4	Average of Day 2 Through 6
7	23			19.2	Average of Day 3 Through 7
8	22			21.4	Average of Day 4 Through 8
9	22			22.4	Average of Day 5 Through 9
10	14	18.1	Average of Day 1 Through 10	21.4	Average of Day 6 Through 10
11	17	18.5	Average of Day 2 Through 11	19.6	Average of Day 7 Through 11
12	19	19	Average of Day 3 Through 12	18.8	Average of Day 8 Through 12
13	13	19.2	Average of Day 4 Through 13	17	Average of Day 9 Through 13
14	16	19.1	Average of Day 5 Through 14	15.8	Average of Day 10 Through 14
15	17	18.9	Average of Day 6 Through 15	16.4	Average of Day 11 Through 15

As you can see, moving average uses the past data to smoothen the price curve. For the purpose of the project, we used 20 day and 50 day moving averages.

Now that we know about the concept of moving average, we will be wondering how to use it to determine whether to buy or sell a stock.

When the shorter-term moving average crosses above the longer-term moving average, it is a signal to **BUY**, as it indicates that the trend is shifting up. This is known as a Golden Cross.

On the opposite when the shorter term moving average crosses below the longer term moving average, it is a signal to **SELL**, as it indicates the trend is shifting down. It is sometimes referred to as the Death Cross.

Please note that it is important that the moving averages cross each other in order to generate a signal. Merely being above or below is not sufficient to generate a signal.

When the signal is neither **buy** nor **sell**, it is classified as **hold**. If you already own the stock, keep it and if you don't then don't buy it now.

So, now that we are clear on Moving average , and have no doubt of whatsoever was happening above , we can freely answer the important questions.

HOW IS THIS HELPFUL TO ANYBODY?

So, to put it simply without data you cant know what the company is doing or whats the value that should be invested and all. And without the analysis of the data which comes out from the company and its stats , one cannot determine how the company is doing or is the company worth investing someone’s money. If you don’t know anything about the company , how will you risk speding your money on buying stocks of that company. Same goes for selling also, for example, you own a company’s stock but now you want to sell it but you don’t know anything about whats going on with the company or the current trend of that company , you will not just sell it for a loss without even checking whether its market going up in the near future or not.

So , that’s all about how its going to help others but what about us , the learners.

Well as per the word above,“learners” , it will help us to learn how to analyze data and more the real life data sets the more you learn. Just theory wouldn’t help when it comes to job or something more complex than the theory. By practising more and more real life data sets we will get to know and get the knowledge to analyze datasets. As we know every dataset has a different structure on how to analyze it. The more you solve different data sets , the more you get the grasp of analyzing different types of datasets.

WORK PLAN AND IMPLEMENTATION OF THE PROJECT

The project of "**STOCK MARKET ANALYSIS"** was assigned by UpGrad which was started in the middle of the course. The project was itself was a part of the course which was very effective for me as it led me to challenge myself in the middle of the course and doing the contents of the course at the same time.

The work plan was simple:

WORKPLAN

DATE	WORK DONE
June 22 , 2021	Planning on how to do project
June 23 , 2021	Research for the project and Starting the project
June 24, 2021	Task 1 and Task 2 of the project done
June 25, 2021	Task 3 done
June 26,2021	Task 4 and Task 5 done
June 27, 2021	Analyzing the outcomes to make the conclusions on the stocks to be invested
June 27, 2021	Project Done

That was the workplan but how I implemented my project is more important.

I used MySQL to complete this project, all codes were implemented on the MySQL workspace only. A bit about MySQL should be mentioned.

MySQL:

MySQL is an open-source relational database management system (RDBMS). Its name is a combination of "My", the name of co-founder Michael Widenius's daughter, and "SQL", the abbreviation for Structured Query Language. A relational database organizes data into one or more data tables in which data types may be related to each other; these relations help structure the data. SQL is a language programmers use to create, modify, and extract data from the relational database, as well as control user access to the database. In addition to relational databases and SQL, an RDBMS like MySQL works with an operating system to implement a relational database in a computer's storage system, manages users, allows for network access, and facilitates testing database integrity and creation of backups.

MySQL is free and open-source software under the terms of the GNU General Public License and is also available under a variety of proprietary licenses. MySQL was owned and sponsored by the Swedish company MySQL AB, which was bought by Sun Microsystems (now Oracle Corporation). In 2010, when Oracle acquired Sun, Widenius forked the open-source MySQL project to create MariaDB.

MySQL has stand-alone clients that allow users to interact directly with a MySQL database using SQL, but more often, MySQL is used with other programs to implement applications that need relational database capability. MySQL is a component of the LAMP web application software stack (and others), which is an acronym for Linux, Apache, MySQL, Perl/PHP/Python. MySQL is used by many database-driven web applications, including Drupal, Joomla, phpBB, and WordPress. MySQL is also used by many popular websites, including Facebook, Flickr, MediaWiki, Twitter, and YouTube.

Now that we know a little bit about MySQL, we can straight jump to how the coding, and all were implemented on the platform.

Below are the snippets of the codes:

SETTING UP THE WORKBENCH AND UPDATING THE TABLES

```
1   -- Creating and Using the assignment Schema and allowing updates
2 • create schema `assignment`;
3 • use assignment;
4 • SET SQL_SAFE_UPDATES = 0;
5
6   -- Imported csv files with the help of table data import wizard
7
8   -- Checking created tables after importing from csv
9 • show tables;
10
11  -- Checking structure of created table
12 • desc bajaj_auto;
13 • desc eicher_motors;
14 • desc hero_motocorp;
15 • desc infosys;
16 • desc tcs;
17 • desc tvs_motors;
```

```sql
-- Checking Data stored in table
select * from bajaj_auto;
select * from eicher_motors;
select * from hero_motocorp;
select * from infosys;
select * from tcs;
select * from tvs_motors;

-- Function to get  date as number
DELIMITER $$
create function get_date(file_date varchar(20))
returns  varchar(2) deterministic
BEGIN
declare format_date varchar(2);
if file_date='January' then
    set format_date= '1';
elseif  file_date='February' then
    set format_date= '2';
elseif file_date='March' then
    set format_date= '3';
elseif file_date='April' then
    set format_date= '4';
elseif  file_date='May' then
    set format_date= '5';
elseif file_date='June' then
    set format_date= '6';
```

```
elseif file_date='July' then
    set format_date= '7';
elseif  file_date='August' then
    set format_date= '8';
elseif file_date='September' then
    set format_date='9';
elseif file_date='October' then
    set format_date='10';
elseif  file_date='November' then
    set format_date= '11';
else
    set format_date= '12';
end if;
return format_date;
END $$
DELIMITER;
-- Function to get Month name from Date
DELIMITER $$
create function getmonth(date varchar(20))
returns varchar(20) deterministic
BEGIN
return (select SUBSTRING_INDEX(SUBSTRING_INDEX(Date,'-',2),'-',-1));
END $$
DELIMITER;

-- Updating Values of Dates according to given format
update bajaj_auto
set Date=(select replace(Date,getmonth(Date),get_date(getmonth(Date))));

update eicher_motors
set Date=(select replace(Date,getmonth(Date),get_date(getmonth(Date))));

update hero_motocorp
set Date=(select replace(Date,getmonth(Date),get_date(getmonth(Date))));

update infosys
set Date=(select replace(Date,getmonth(Date),get_date(getmonth(Date))));

update tcs
set Date=(select replace(Date,getmonth(Date),get_date(getmonth(Date))));

 update tvs_motors
set Date=(select replace(Date,getmonth(Date),get_date(getmonth(Date))));
```

```sql
-- Update Values of Dates to be taken with str_to_date function and convert to date

update bajaj_auto
set `Date`=  str_to_date(`Date`,'%d-%m-%Y');
alter table bajaj_auto
modify `Date` date ;

update eicher_motors
set `Date`=  str_to_date(`Date`,'%d-%m-%Y');
alter table eicher_motors
modify `Date` date ;

update hero_motocorp
set `Date`=  str_to_date(`Date`,'%d-%m-%Y');
alter table hero_motocorp
modify `Date` date ;


update infosys
set `Date`=  str_to_date(`Date`,'%d-%m-%Y');
alter table infosys
modify `Date` date ;

update tcs
set `Date`=  str_to_date(`Date`,'%d-%m-%Y');
alter table tcs
modify `Date` date ;

update tvs_motors
set `Date`=  str_to_date(`Date`,'%d-%m-%Y');
alter table tvs_motors
modify `Date` date ;
```

STARTING WITH TASK 1

```
-- TASK 1

-- Create table bajaj1
create table bajaj1 as
    select `Date`,`Close Price`,
avg(`Close Price`) over(order by `Date` rows between 19 preceding and current row) as '20 Day MA',
avg(`Close Price`) over(order by `Date` rows between 49 preceding and current row) as '50 Day MA'
 from bajaj_auto;

 -- Create table eicher1
 create table eicher1 as
    select `Date`,`Close Price`,
avg(`Close Price`) over(order by `Date` rows between 19 preceding and current row) as '20 Day MA',
avg(`Close Price`) over(order by `Date` rows between 49 preceding and current row) as '50 Day MA'
 from eicher_motors;

-- Create table hero1
create table hero1 as
   select `Date`,`Close Price`,
avg(`Close Price`) over(order by `Date` rows between 19 preceding and current row) as '20 Day MA',
avg(`Close Price`) over(order by `Date` rows between 49 preceding and current row) as '50 Day MA'
from hero_motocorp;

-- Create table infosys1
Create table infosys1
select `Date`,`Close price`,
avg(`Close price`) over(order by `Date` rows  between 19 preceding and current row) as '20 Day MA',
avg(`Close price`) over(order by `Date` rows  between 49 preceding and current row) as '50 Day MA'
from infosys;

 -- Create table tcs1
 Create table tcs1
 select `Date`,`Close price`,
 avg(`Close price`) over(order by `Date` rows  between 19 preceding and current row) as '20 Day MA',
 avg(`Close price`) over(order by `Date` rows  between 49 preceding and current row) as '50 Day MA'
 from tcs;

 -- Create table tvs1
 Create table tvs1
 select `Date`,`Close price`,
 avg(`Close price`) over(order by `Date` rows  between 19 preceding and current row) as '20 Day MA',
 avg(`Close price`) over(order by `Date` rows  between 49 preceding and current row) as '50 Day MA'
 from tvs_motors;
```

```sql
-- Making first 19 rows NULL as moving average can't be calculated
update bajaj1
set `20 Day MA` = NULL limit 19;

update eicher1
set `20 Day MA` = NULL limit 19;

update hero1
set `20 Day MA` = NULL limit 19;

update infosys1
set `20 Day MA` = NULL limit 19;

update tcs1
set `20 Day MA` = NULL limit 19;

update tvs1
set `20 Day MA` = NULL limit 19;


-- Making first 49 rows NULL as moving average can't be calculated
update bajaj1
set `50 Day MA` = NULL limit 49;

update eicher1
set `50 Day MA` = NULL limit 49;

update hero1
set `50 Day MA` = NULL limit 49;

update infosys1
set `50 Day MA` = NULL limit 49;

update tcs1
set `50 Day MA` = NULL limit 49;

update tvs1
set `50 Day MA` = NULL limit 49;

-- Checking Data
select * from bajaj1;
select * from eicher1;
select * from hero1;
select * from infosys1;
select * from tcs1;
select * from tvs1;
```

TASK 2

```
-- TASK 2

-- Create master_stock_info table

create table master_stock_info
select tcs.`Date`,b.`Close price` as 'Bajaj',
tcs.`Close price` as 'TCS' ,tvs.`Close price` as 'TVS',
i.`Close price` as 'Infosys',e.`Close price` as 'Eicher',
h.`Close price` as 'Hero'
from tcs  inner join eicher_motors e on e.`Date`=tcs.`Date`
inner join  tvs_motors tvs on tvs.`Date`= tcs.`Date`
inner join  hero_motocorp h on h.`Date` = tcs.`Date`
inner join  bajaj_auto b on b.`Date`=tcs.`Date`
inner join  infosys i on i.`Date`=tcs.`Date` order by tcs.`Date`;

-- Display data from master_stock_info
select * from master_stock_info;
```

TASK 3

```
-- TASK 3

-- create table bajaj2
create table bajaj2
select `Date`,`Close price`,
case
when `50 Day MA` is NULL then 'NA'
when `20 Day MA`>`50 Day MA` and ((lag(`20 Day MA`,1) over(order by `Date`))<(lag(`50 Day MA`,1) over(order by `Date`))) then 'BUY'
when `20 Day MA`<`50 Day MA` and ((lag(`20 Day MA`,1) over(order by `Date`))>(lag(`50 Day MA`,1) over(order by `Date`))) then 'SELL'
else 'HOLD'
end as `Signal`
from bajaj1 ;

-- create table eicher2
create table eicher2
select `Date`,`Close price`,
case
when `50 Day MA` is NULL then 'NA'
when `20 Day MA`>`50 Day MA` and ((lag(`20 Day MA`,1) over(order by `Date`))<(lag(`50 Day MA`,1) over(order by `Date`))) then 'BUY'
when `20 Day MA`<`50 Day MA` and ((lag(`20 Day MA`,1) over(order by `Date`))>(lag(`50 Day MA`,1) over(order by `Date`))) then 'SELL'
else 'HOLD'
end as `Signal`
from eicher1;
```

```
-- create table tcs2
create table tcs2
select `Date`,`Close price`,
case
when `50 Day MA` is NULL then 'NA'
when `20 Day MA`>`50 Day MA` and ((lag(`20 Day MA`,1) over(order by `Date`))<(lag(`50 Day MA`,1) over(order by `Date`))) then 'BUY'
when `20 Day MA`<`50 Day MA` and ((lag(`20 Day MA`,1) over(order by `Date`))>(lag(`50 Day MA`,1) over(order by `Date`))) then 'SELL'
else 'HOLD'
end as `Signal`
from tcs1 ;

-- create table tvs2
create table tvs2
select `Date`,`Close price`,
case
when `50 Day MA` is NULL then 'NA'
when `20 Day MA`>`50 Day MA` and ((lag(`20 Day MA`,1) over(order by `Date`))<(lag(`50 Day MA`,1) over(order by `Date`))) then 'BUY'
when `20 Day MA`<`50 Day MA` and ((lag(`20 Day MA`,1) over(order by `Date`))>(lag(`50 Day MA`,1) over(order by `Date`))) then 'SELL'
else 'HOLD'
end as `Signal`
from tvs1 ;

-- create table hero2
create table hero2
select `Date`,`Close price`,
case
when `50 Day MA` is NULL then 'NA'
when `20 Day MA`>`50 Day MA` and ((lag(`20 Day MA`,1) over(order by `Date`))<(lag(`50 Day MA`,1) over(order by `Date`))) then 'BUY'
when `20 Day MA`<`50 Day MA` and ((lag(`20 Day MA`,1) over(order by `Date`))>(lag(`50 Day MA`,1) over(order by `Date`))) then 'SELL'
else 'HOLD'
end as `Signal`
from hero1 ;

-- create  table infosys2
create table infosys2
select `Date`,`Close price`,
case
when `50 Day MA` is NULL then 'NA'
when `20 Day MA`>`50 Day MA` and ((lag(`20 Day MA`,1) over(order by `Date`))<(lag(`50 Day MA`,1) over(order by `Date`))) then 'BUY'
when `20 Day MA`<`50 Day MA` and ((lag(`20 Day MA`,1) over(order by `Date`))>(lag(`50 Day MA`,1) over(order by `Date`))) then 'SELL'
else 'HOLD'
end as `Signal`
from infosys1 ;

-- Checking Data

select * from bajaj2;
select * from eicher2;
select * from hero2;
select * from infosys2;
select * from tcs2;
select * from tvs2;
```

TASK 4

```
-- TASK 4

DELIMITER $$
create function get_signal(signal_date date)
returns varchar(20) deterministic
BEGIN
return (select `Signal` from bajaj2 where bajaj2.`Date` = signal_date);
END $$
DELIMITER ;
-- testing of function
select get_signal(`Date`),`Date` as 'Signal' from bajaj_auto;
select get_signal('2018-06-21');
-- Expected output  BUY , actual output BUY ->Pass
select get_signal('2018-05-29');
-- Expected output SELL, actual output SELL ->Pass
select get_signal('2018-05-30');
-- Expected output HOLD, actual output HOLD ->Pass
select get_signal('2015-01-01');
-- Expected output NA, actual output NA ->Pass
```

TASK 5

```
-- Task 5 analysis


-- Getting the number of times bought and sold

select count(*) from bajaj2 where `Signal`='SELL';
select count(*) from bajaj2 where `Signal`='BUY';

select * from bajaj2 where `Signal`='BUY' or `Signal`='SELL';

select * from tcs2 where `Signal`='BUY' or `Signal`='SELL';
select count(*) from tcs2 where `Signal`='SELL';

select * from eicher2 where `Signal`='BUY' or `Signal`='SELL';
select count(*) from eicher2 where `Signal`='SELL';
select count(*) from eicher2 where `Signal`='BUY';

select * from tvs2 where `Signal`='BUY' or `Signal`='SELL';
select count(*) from tvs2 where `Signal`='SELL';
select count(*) from tvs2 where `Signal`='BUY';

select * from hero2 where `Signal`='BUY' or `Signal`='SELL';
select count(*) from hero2 where `Signal`='SELL';
select count(*) from hero2 where `Signal`='BUY';
```

```
select * from infosys2 where `Signal`='BUY' or `Signal`='SELL';
select count(*) from infosys2 where `Signal`='SELL';
select count(*) from infosys2 where `Signal`='BUY';

 -- Getting the trend
 select (select `Close price` from bajaj_auto  order by `Date` desc limit 1) - (select `Close price` from bajaj_auto  order by `Date`  limit 1) as 'Trend';

 select (select `Close price` from tcs  order by `Date` desc limit 1) - (select `Close price` from tcs  order by `Date`  limit 1) as 'Trend';

 select (select `Close price` from eicher_motors  order by `Date` desc limit 1) - (select `Close price` from eicher_motors  order by `Date`  limit 1) as 'Trend';

 select (select `Close price` from tvs_motors  order by `Date` desc limit 1) - (select `Close price` from tvs_motors  order by `Date`  limit 1) as 'Trend';

 select (select `Close price` from hero_motocorp  order by `Date` desc limit 1) - (select `Close price` from hero_motocorp  order by `Date`  limit 1) as 'Trend';

 select (select `Close price` from infosys  order by `Date` desc limit 1) - (select `Close price` from infosys  order by `Date`  limit 1) as 'Trend';
```

After the implementation of the above codes, we can now make the conclusions on determining the status of the stocks of each of the companies.

Here is a report of the above observation:

KEY OBSERVATIONS

Bajaj Auto

- Bajaj Auto Stock could have been bought 12 times, sold 11 times in the provided time.
- Bajaj Auto Stock's trend as of 21-06-2018 is shifting up.
- Date: 31-07-2018 = 2700.7, 01-01-2015 = 2454.1, results in 10% increase in stock value.

TCS

- TCS Stock could have been bought 12 times, sold 13 times in the provided time.
- TCS Stock's Trend as of 05-06-2018 is shifting down.
- Date: 31-07-2018 =1941.25, 01-01-2015 = 2548.2, results in 23.8% decrease in stock value.

Eicher Motors

- Eicher's Stock could have been bought 6 times, sold 7 times in the provided time.
- Eicher's Stock Trend as of 06-06-2018 is shifting down.
- Date: 31-07-2018= 27820.95, 01-01-2015=15239.15, results in 82.5% increase in stock value.

TVS Motors

▪ TVS Stock could have been bought 8 times, sold 8 times in the provided time.

▪ TVS Stock Trend as of 17-05-2018 is shifting down.

▪ Date: 31-07-2018= 517.45, 01-01-2015= 276.85, results in 86.9% increase in stock value.

Hero Motocorp

▪ Hero Motocorp Stock could have been bought 9 times, sold 9 times in the provided time.

▪ Hero Motocorp Stock Trend as of 22-05-2018 is shifting down.

▪ Date: 31-07-2018= 3293.8, 01-01-2015= 3107.3, results in 6% increase in stock value.

Infosys

▪ Infosys Stock could have been bought 9 times, sold 9 times in the provided time.

▪ Infosys Stock Trend as of 07-05-2018 is shifting up.

▪ Date: 31-07-2018: 1365, 01-01-2015: 1975.8, results in 3% decrease in stock value.

Conclusion from above:

• Stocks that can be bought in likelihood of returns are Bajaj Auto, Infosys.

• Stocks that can be sold in likelihood of returns are Eicher Motors, TVS Motors, Hero Motocorp, TCS.

With this we can basically conclude all about my project.

Now from the next chapter, we will learn about the experience I gained from the summer training course on **“BASIC OF DATA SCIENCE”** and the learning outcomes and how will it help me in my career.

CHAPTER – II

INTRODUCTION TO DATA SCIENCE

Data science is a multidisciplinary approach to extracting actionable insights from the large and ever-increasing volumes of data collected and created by today's organizations. Data science encompasses preparing data for analysis and processing, performing advanced data analysis, and presenting the results to reveal patterns and enable stakeholders to draw informed conclusions.

Data preparation can involve cleansing, aggregating, and manipulating it to be ready for specific types of processing. Analysis requires the development and use of algorithms, analytics and AI models. It's driven by software that combs through data to find patterns within to transform these patterns into predictions that support business decision-making. The accuracy of these predictions must be validated through scientifically designed tests and experiments. And the results should be shared through the skilful use of data visualization tools that make it possible for anyone to see the patterns and understand trends.

As a result, data scientists (as data science practitioners are called) require computer science and pure science skills beyond those of a typical data analyst. A data scientist must be able to do the following:

- Apply mathematics, statistics, and the scientific method
- Use a wide range of tools and techniques for evaluating and preparing data—everything from SQL to data mining to data integration methods
- Extract insights from data using predictive analytics and artificial intelligence (AI), including machine learning and deep learning models
- Write applications that automate data processing and calculations
- Tell—and illustrate—stories that clearly convey the meaning of results to decision-makers and stakeholders at every level of technical knowledge and understanding
- Explain how these results can be used to solve business problems

DATA SCIENCE TOOLS:

Data scientists must be able to build and run code to create models. The most popular programming languages among data scientists are open-source tools that include or support pre-built statistical, machine learning and graphics capabilities. These languages include:

1. **R**
2. **Python**

What is R?

R is an open-source programming language that's optimized for statistical analysis and data visualization. Developed in 1992, R has a rich ecosystem with complex data models and elegant tools for data reporting. At last count, more than 13,000 R packages were available via the Comprehensive R Archive Network (CRAN) for deep analytics.

Popular among data science scholars and researchers, R provides a broad variety of libraries and tools for the following:

- Cleansing and prepping data
- Creating visualizations
- Training and evaluating machine learning and deep learning algorithms

R is commonly used within RStudio, an integrated development environment (IDE) for simplified statistical analysis, visualization, and reporting. R applications can be used directly and interactively on the web via Shiny.

What is Python?

Python is a general-purpose, object-oriented programming language that emphasizes code readability through its generous use of white space. Released in 1989, Python is easy to learn and a favorite of programmers and developers. In fact, Python is one of the most popular programming languages in the world, just behind Java and C.

Several Python libraries support data science tasks, including the following:

- Numpy for handling large dimensional arrays
- Pandas for data manipulation and analysis
- Matplotlib for building data visualization

Plus, Python is particularly well suited for deploying machine learning at a large scale. Its suite of specialized deep learning and machine learning libraries includes tools like scikit-learn, Keras and TensorFlow, which enable data scientists to develop sophisticated data models that plug directly into a production system. Then, Jupyter Notebooks are an open source web application for easily sharing documents that contain your live Python code, equations, visualizations and data science explanations.

CHAPTER III
TECHNOLOGY LEARNT

DATA ANALYSIS USING EXEL:

Microsoft Excel is a spreadsheet program used to record and analyze numerical and statistical data. Microsoft Excel provides multiple features to perform various operations like calculations, pivot tables, graph tools, macro programming, etc. It is compatible with multiple OS like Windows, macOS, Android and iOS.

An Excel spreadsheet can be understood as a collection of columns and rows that form a table. Alphabetical letters are usually assigned to columns, and numbers are usually assigned to rows. The point where a column and a row meet, is called a cell. The address of a cell is given by the letter representing the column and the number representing a row.

Navigating through data could be a nightmare in itself.

It's quite tricky to explore and process data when you are looking at large chunks of data. Analysing it could very well be a unique challenge. However, Excel can come to your rescue.

Excel contains functions that can process a large amount of data quite effectively and easily. While different tasks of data analysis could be tricky, Excel functions are quite easy, and anybody can use them and analyse the data.

It's not necessary either to remember all the functions. You can simply Google it and find out the function you need for data analysis tasks.

For the sheer speed, simplicity and accuracy of it, Excel is not just useful but imperative for data analysis. It can save your valuable time and effectively enable the data analysis without any hassle as well.

When it comes to data analysis with Excel, there are four major steps or processes:

- Data collection
- Data Cleaning
- Data Exploration (using Pivot Table)
- Data Visualization

1.Data Collection:

In order to get started with data analysis, the first step is to collect information on the variables in a systematic way. This kind of a process will help us find answers to the important questions and assess the results.

Data collection part is vital because it ensures the accuracy of the data so that decisions related to the data turn out to be valid.

Data collection is also useful because you have a baseline with which you can measure, and you also get a target where you aim at reaching.

As regards Excel, it is possible for you to collect and import data from a diversity of data sources. Your data sources could be:

- Web Page
- Microsoft Access Database

2.Data Cleaning:

Data cleaning is all about finding out and correcting the errors in the dataset. It also includes replacing the incomplete or inaccurate parts with the correct ones.

In Excel, you can clean data by using the techniques given below:

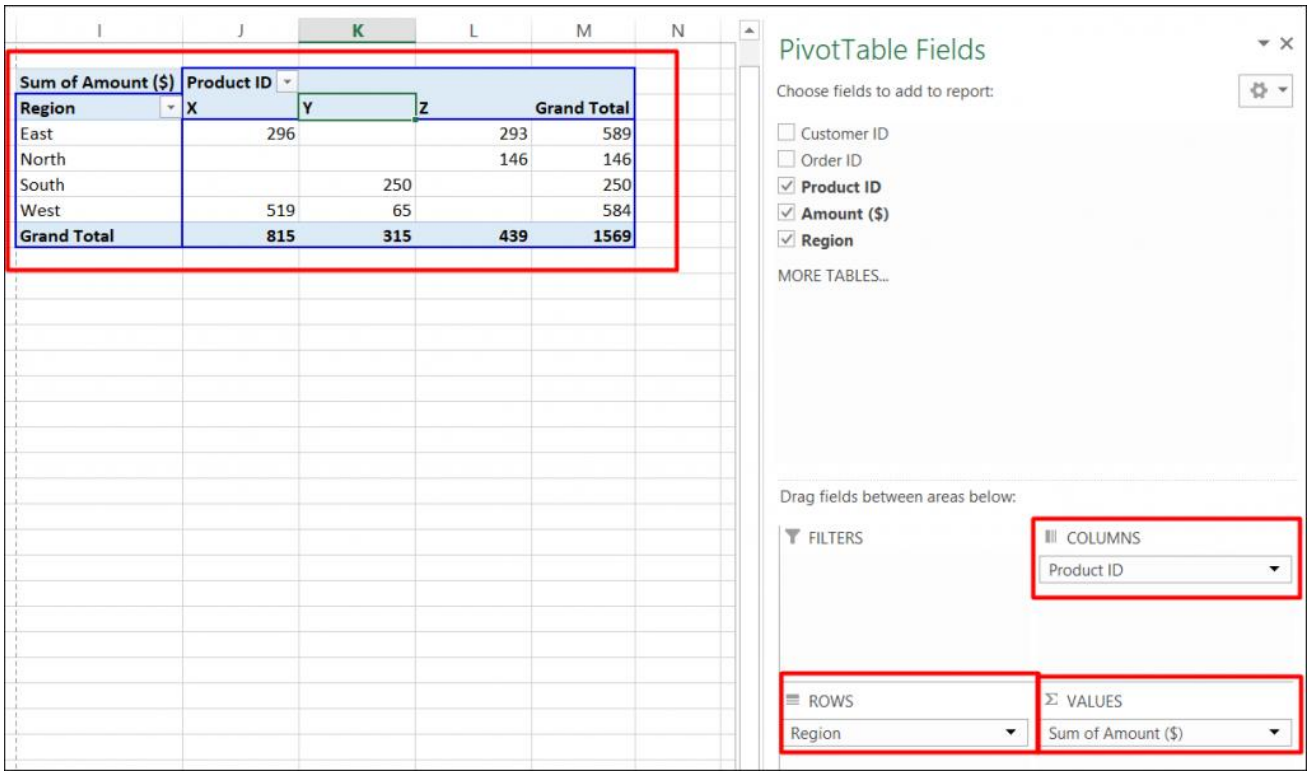
- Removing duplicate values
- Removing spaces
- Merging and splitting columns
- Reconciling table data by joining or matching

3.Data Exploration:

Data Exploring is the vital process of performing initial investigations on data in order to find out patterns, to spot anomalies, to test hypothesis and to check assumptions with the help of summary statistics and graphical representations.

Why it matters so much is that you can make use of exploring data and make sense of the data you have. You can then figure out what questions you want to ask and how to frame them, as well as how best to manipulate your available data sources to get the answers you need.

Pivot Table- Excel's Pivot Table is a summary table that lets you count, average, sum, and perform other calculations according to the reference feature you have selected.



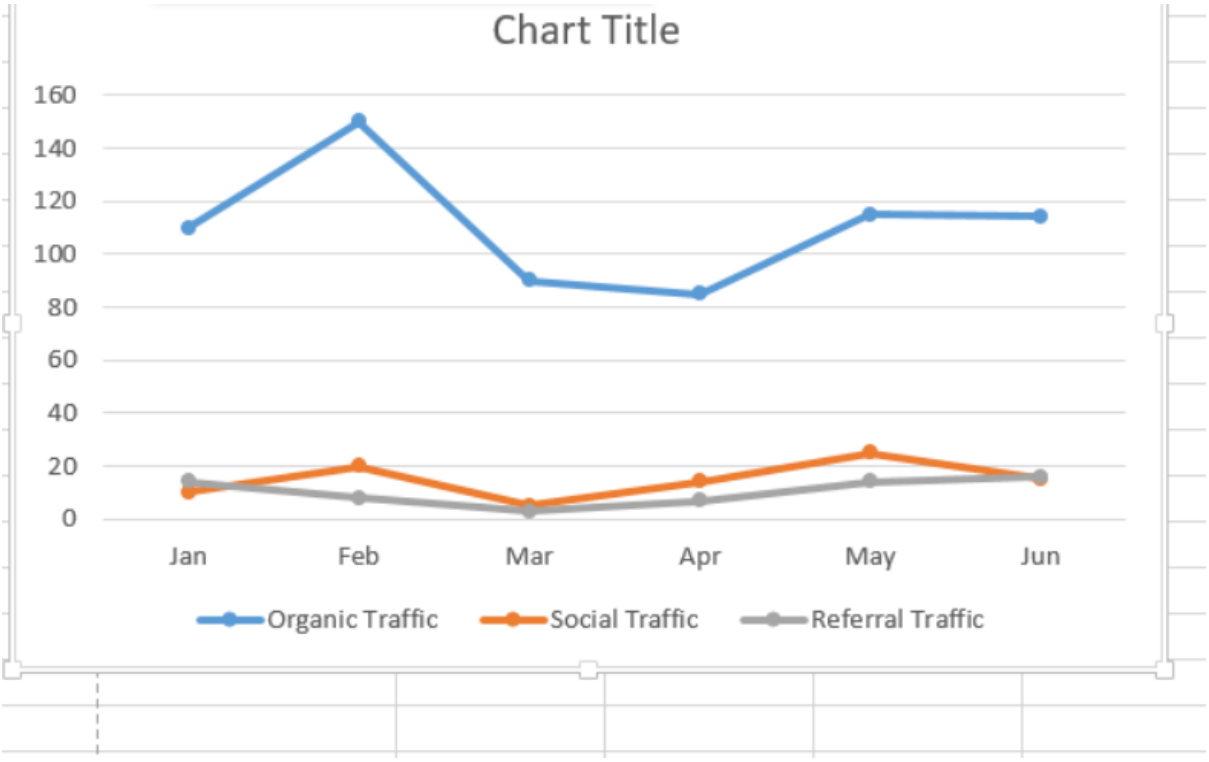
4.Data Visualisation:

As exploring data is quite important, data visualization as a technique through which we can explore data also becomes vital for us.

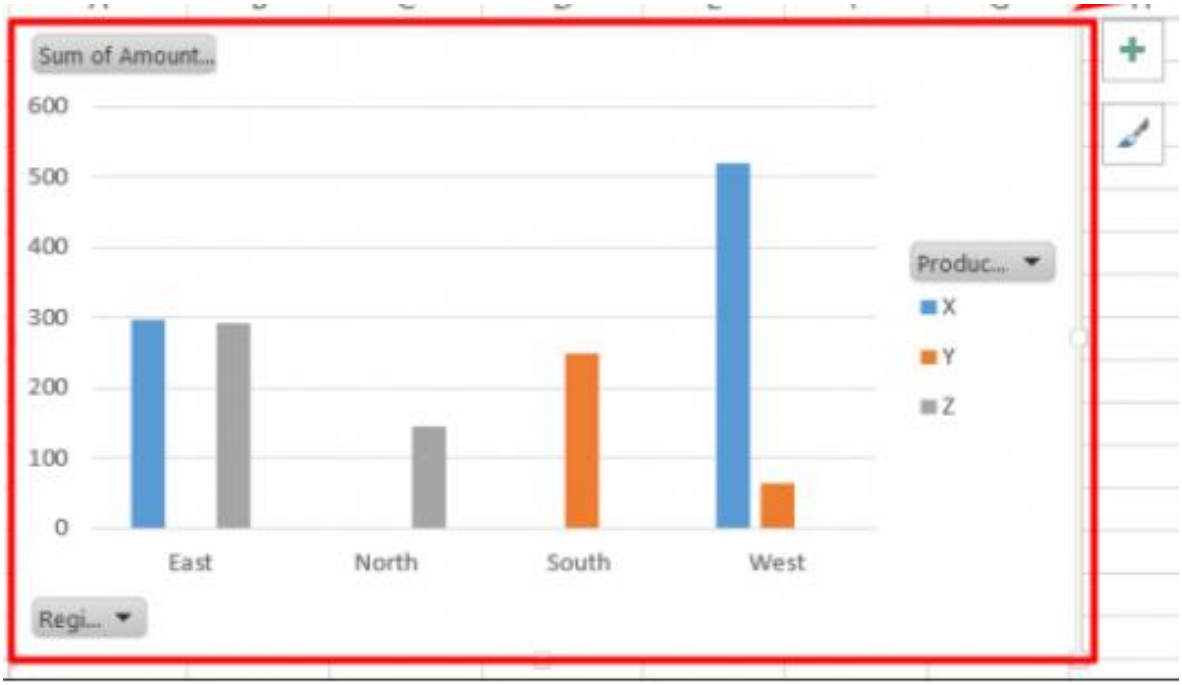
Data visualization is the presentation of data in a pictorial or graphical format. The reason why such a graphical format matters is that it becomes easier for decision makers to see analytics presented visually. In other words, they can grasp difficult concepts or identify new patterns far more easily.

In Excel, there are 2 features (**Charts and Pivot Charts**) which are most popular for data visualization.

Charts- A simple chart in Excel can say a lot more than a sheet full of numbers.



Pivot Chart - A pivot chart is the visual representation of a pivot table in Excel. Pivot charts and pivot tables are connected with each other.



DATA ANALYSIS USING PYTHON:

Python is a very flexible and general-purpose language that during the years has gained more and more credit in the data analysis community. Unlike other languages, such as R, Scala, Matlab or Julia, Python has not been conceived to perform data analysis and in general scientific and numerical tasks, but this may be considered an advantage, because with Python we can do just…anything.

Stats show that in 2020 around 66% of data scientists are using Python on a daily basis and 84% use it as their main language. It is also worth noting that around Python a huge and very active community has developed so if you have a problem or want to collaborate, it's quite simple to find someone to work with. But how do you perform data analysis in Python?

With the help of some essential libraries, we can analyse data very effectively.

You should think of libraries as a set of tools ready to use that someone else developed to make certain coding tasks easier. So instead of having the burden to build a function that performs a certain operation, you can simply go to a library and just use an already made function. The wonderful thing about Python is that since it is so diffused and so widespread into the data analysis community there are really powerful dedicated libraries that you can use for your data analysis problems. Furthermore, there is a lot of documentation for each library. The main libraries for data science are:

- **NUMPY:**

  Numpy, which stands for "numerical python", is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays.It offers pre-compiled functions for numerical routines.

- **PANDAS:**

  This is perfect for data analysis, manipulation and visualisation. It allows high-level data structures and some tools to manipulate them. It offers data structures and operations for manipulating numerical tables and time series. It is free software released under the three-clause BSD license. The name is derived from the term "panel data", an econometrics term for data sets that include observations over multiple time periods for the same individuals. Its name is a play on the phrase "Python data analysis" itself.

- **MATPLOTLIB:**

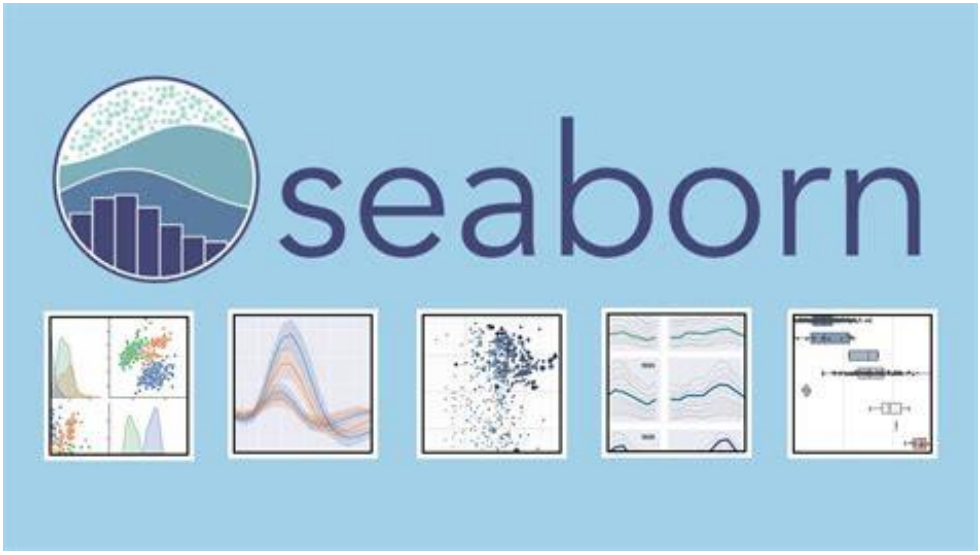
  Matplotlib is one of the most popular Python packages used for data visualization. It is a cross-platform library for making 2D plots from data in arrays. It provides an object-oriented API that helps in embedding plots in applications using Python GUI toolkits such as PyQt, WxPythonotTkinter. It can be used in Python and IPython shells, Jupyter notebook and web application servers also.

- **SCIPY:**

  SciPy is a scientific computation library that uses NumPy underneath. SciPy stands for Scientific Python. It provides more utility functions for optimization, stats and signal processing. Just like NumPy, SciPy is open source so we can use it freely. SciPy has optimized and added functions that are frequently used in NumPy and Data Science. Basically, Scipy is for algebra, statistics, linear algebra.

- **SEABORN:**

  Seaborn is a library for making statistical graphics in Python. It builds on top of matplotlib and integrates closely with pandas data structures.
  Seaborn helps you explore and understand your data. Its plotting functions operate on data frames and arrays containing whole datasets and internally perform the necessary semantic mapping and statistical aggregation to produce informative plots. Its dataset-oriented, declarative API lets you focus on what the different elements of your plots mean, rather than on the details of how to draw them.



DATA ANALYSIS USING SQL:

Structured Query Language (SQL) has been around for decades. It is a programming language used for managing the data held in relational databases. SQL is used all around the world by a majority of big companies. A data analyst can use SQL to access, read, manipulate, and analyze the data stored in a database and generate useful insights to drive an informed decision-making process.

There are 8 important SQL techniques for data analysis:

- SQL Technique #1: Counting Rows and Items
- SQL Technique #2: Aggregation Functions
- SQL Technique #3: Extreme Value Identification
- SQL Technique #4: Slicing Data
- SQL Technique #5: Limiting Data
- SQL Technique #6: Sorting Data
- SQL Technique #7: Filtering Patterns
- SQL Technique #8: Groupings, Rolling up Data and Filtering in Groups

SQL Technique #1: Counting Rows and Items-

- **Count Function** –

  The SQL COUNT function is an aggregate function that returns the number of rows returned by a query. For example,

```
mysql> select count(*) from ConsumerDetails;
+----------+
| count(*) |
+----------+
|       10 |
+----------+
1 row in set (0.00 sec)
```

- **Distinct Function –**
  The SELECT DISTINCT statement is used to return only distinct (different) values.

```
SELECT DISTINCT column1, column2, ...
FROM table_name;
```

SQL Technique #2 – Aggregation Functions-

- **SUM ()**

  We use the SUM () function to calculate the sum of the numerical column in a table.

```
mysql> select sum(Total_amt_spend) as sum_all from ConsumerDetails ;
+---------+
| sum_all |
+---------+
|   12560 |
+---------+
1 row in set (0.00 sec)
```

- **AVG ()**

  To calculate the average of the numeric columns, we use the AVG () function.

```
mysql> select avg(Total_amt_spend) as avg_total_spend from ConsumerDetails;
+-----------------+
| avg_total_spend |
+-----------------+
|       1256.0000 |
+-----------------+
1 row in set (0.00 sec)
```

- **STDDEV ()**

  It is used to calculate the Standard Deviation of total records (or rows) selected by the SELECT Statement.

```
mysql> select stddev(Total_amt_spend) as std_total_spend from ConsumerDetails;
+-------------------+
| std_total_spend   |
+-------------------+
| 829.7132034624976 |
+-------------------+
1 row in set (0.00 sec)
```

SQL Technique #3 – Extreme Value Identification

- **MAX ()**

  The maximum numeric value can be identified by using the MAX () function.

```
mysql> select max(Total_amt_spend) as max_spend from ConsumerDetails;
+-----------+
| max_spend |
+-----------+
|      3000 |
+-----------+
1 row in set (0.00 sec)
```

- **MIN ()**

  We have the MIN () function to identify the minimum numeric value in a given column.

```
mysql> select min(Total_amt_spend) as min_spend from ConsumerDetails;
+-----------+
| min_spend |
+-----------+
|       350 |
+-----------+
1 row in set (0.00 sec)
```

SQL Technique #4: Slicing Data

Now, let us focus on one of the most important parts of the data analysis – slicing the data. This section of the analysis is going to form the basis for advanced queries and help you retrieve data based on some kind of condition.

- Let's say that the retail store wants to find the customers coming from a locality, specifically Shakti Nagar and Shanti Vihar. The code for this would be :

```
mysql> select Name from ConsumerDetails where Locality in ("Shakti Nagar", "Shanti Vihar");
+----------+
| Name     |
+----------+
| Shantanu |
| Natasha  |
| Kapil    |
+----------+
3 rows in set (0.00 sec)
```

We have used the WHERE clause to filter out the data based on the condition that consumers should be living in the locality – Shakti Nagar and Shanti Vihar. I didn't use the OR condition here. Instead, I have used the IN operator which allows us to specify multiple values in the WHERE clause.

- This time the retail store wants to retrieve all the consumers who are spending between Rs. 1000 and Rs. 2000 so as to push out special marketing offers. The query for this would be

```
mysql> select Name from ConsumerDetails where Total_amt_spend between 1000 and 2000;
+-------+
| Name  |
+-------+
| Rohan |
+-------+
1 row in set (0.00 sec)
```

SQL Technique #5: Limiting Data

- **Limit**

  The LIMIT clause is used to set an upper limit on the number of tuples returned by SQL.

```
mysql> select * from ConsumerDetails limit 5;
+-------+--------------+------------------+---------------+
| Name  | Locality     | Total_amt_spend | Industry      |
+-------+--------------+------------------+---------------+
| Raj   | Raj Nagar    |             750 | Manufacturing |
| Ajay  | Vijay Nagar  |             500 | Creative      |
| Sagar | Shivam Nagar |             900 | News          |
| Akul  | Preet Vihar  |             350 | Teaching      |
| Rohan | kakar Vihar  |            1150 | Tech          |
+-------+--------------+------------------+---------------+
5 rows in set (0.00 sec)
```

- **OFFSET**

  The OFFSET clause skips the specified number of rows.

```
mysql> select * from ConsumerDetails limit 2 offset 3 ;
+-------+-------------+-----------------+----------+
| Name  | Locality    | Total_amt_spend | Industry |
+-------+-------------+-----------------+----------+
| Akul  | Preet Vihar |             350 | Teaching |
| Rohan | kakar Vihar |            1150 | Tech     |
+-------+-------------+-----------------+----------+
2 rows in set (0.00 sec)
```

SQL Technique #6 – Sorting Data

- **ORDER BY**

  The keyword can be used to sort the data into ascending or descending order. The ORDER BY keyword sorts the data in ascending order by default.

  Example in ascending order,

```
mysql> select * from ConsumerDetails order by Total_amt_spend;
+----------+-------------+-----------------+---------------+
| Name     | Locality    | Total_amt_spend | Industry      |
+----------+-------------+-----------------+---------------+
| Akul     | Preet Vihar |             350 | Teaching      |
| Ajay     | Vijay Nagar |             500 | Creative      |
| Kapil    | shakti nagar|             700 | Aviation      |
| Raj      | Raj Nagar   |             750 | Manufacturing |
| Sagar    | Shivam Nagar|             900 | News          |
| Tanamy   | sikkim nagar|             900 | Defense       |
| Rohan    | kakar Vihar |            1150 | Tech          |
| Shantanu | Shanti Vihar|            2110 | Defense       |
| Natasha  | shakti nagar|            2200 | Aviation      |
| Tarun    | nikepur     |            3000 | Manufacturing |
+----------+-------------+-----------------+---------------+
10 rows in set (0.00 sec)
```

Example in descending order

```
mysql> select * from ConsumerDetails order by Total_amt_spend desc;
+----------+--------------+-----------------+---------------+
| Name     | Locality     | Total_amt_spend | Industry      |
+----------+--------------+-----------------+---------------+
| Tarun    | nikepur      |            3000 | Manufacturing |
| Natasha  | shakti nagar |            2200 | Aviation      |
| Shantanu | Shanti Vihar |            2110 | Defense       |
| Rohan    | kakar Vihar  |            1150 | Tech          |
| Sagar    | Shivam Nagar |             900 | News          |
| Tanamy   | sikkim nagar |             900 | Defense       |
| Raj      | Raj Nagar    |             750 | Manufacturing |
| Kapil    | shakti nagar |             700 | Aviation      |
| Ajay     | Vijay Nagar  |             500 | Creative      |
| Akul     | Preet Vihar  |             350 | Teaching      |
+----------+--------------+-----------------+---------------+
10 rows in set (0.00 sec)
```

SQL Technique #7 – Filtering Patterns

- **LIKE operator**

  The LIKE operator is used in a WHERE clause to search for a specified pattern in a column.
- **Wildcard Characters**

  The Wildcard Character is used to substitute one or more characters in a string. These are used along with the LIKE operator. The two most common wildcard characters are:

  1. **%,** It represents 0 or more number of characters
  2. **_**, It represents a single character

```
mysql> select * from ConsumerDetails where Locality like "%Nagar";
+---------+--------------+-----------------+---------------+
| Name    | Locality     | Total_amt_spend | Industry      |
+---------+--------------+-----------------+---------------+
| Raj     | Raj Nagar    |             750 | Manufacturing |
| Ajay    | Vijay Nagar  |             500 | Creative      |
| Sagar   | Shivam Nagar |             900 | News          |
| Natasha | shakti nagar |            2200 | Aviation      |
| Kapil   | shakti nagar |             700 | Aviation      |
| Tanamy  | sikkim nagar |             900 | Defense       |
+---------+--------------+-----------------+---------------+
6 rows in set (0.00 sec)
```

SQL Technique #8 – Groupings, Rolling up Data and Filtering in Groups

- **GROUP BY**

  Grouping of data is performed using the GROUP BY statement. The most useful application of this statement is to find the distribution of categorical variables. This is done by using the GROUP BY statement along with aggregation functions like – COUNT, SUM, AVG, etc.

```
mysql> select count(*), Industry from ConsumerDetails group by Industry;
+----------+---------------+
| count(*) | Industry      |
+----------+---------------+
|        2 | Aviation      |
|        1 | Creative      |
|        2 | Defense       |
|        2 | Manufacturing |
|        1 | News          |
|        1 | Teaching      |
|        1 | Tech          |
+----------+---------------+
7 rows in set (0.01 sec)
```

- **HAVING**

  The HAVING clause is just like the WHERE clause but only for filtering the grouped by data. Remember, it will always come after the GROUP BY statement.

```
mysql> select sum(total_amt_spend) as category_sum, Industry from ConsumerDetails group by Industry having category_sum >2500;
+--------------+--------------+
| category_sum | Industry     |
+--------------+--------------+
|         2900 | Aviation     |
|         3010 | Defense      |
|         3750 | Manufacturing |
+--------------+--------------+
3 rows in set (0.00 sec)
```

CHAPTER IV – FINAL CHAPTER

CONCLUSION AND FUTURE PERSPECTIVE

FINDINGS AND KEY TAKEAWAYS FROM THE TRAINING:

- This training helped me to kickstart my career in Data Science.
- The environment and engagements between peers helped to experience the problem-solving environment which was very amazing to experience it.
- The deep knowledge of Exel which was never been known before was enlightened to me.
- The data analysis side of python was introduced to us, and the visualisations were the "cherry topping" to the coding and work done within the course.
- The introduction to SQL was very much crucial to us and this language was the one which led me to have this project and I was impressed by the outcome.
- Basics of Tableau was also taught during the course which was very helpful and now we can later dive into the deep roots of Tableau later.
- Then the final module was on Statistics of Data Science, which consisted of Exploratory Data analysis, Inferential Statistics, and Hypothesis Testing.
- The most lovable part of this training was it was time bounded and it was an interactive course, not being a self-paced test where there is no real entity interaction. Because of this only, there were live session 3 days a week which was super interactive and productive. The instructors were really good with their jobs.
- Another major quality was the practice with real life datasets and the weakly assignments, which never led the students to slack off and be bored.
- This 8-week training was one of the most productive and most learning weeks of my life.

Now, these were my major takeaways and per say review of the whole 8 weeks training.

But the question is why Data Science? And how is it going to help me in my career?

WHY DATA SCIENCE?

Data Science is the study of data. It is about extracting, analyzing, visualizing, managing and storing data to create insights. These insights help the companies to make powerful data-driven decisions. Data Science requires the usage of both unstructured and structured data. It is a multidisciplinary field that has its roots in statistics, math and computer science. It is one of the most highly sought-after jobs due to the abundance of data science position and a lucrative pay-scale.

The various benefits of Data Science are as follows:

1. **It's in Demand**

Data Science is greatly in demand. Prospective job seekers have numerous opportunities. It is the fastest growing job on Linkedin and is predicted to create 11.5 million jobs by 2026. This makes Data Science a highly employable job sector.

2. **Abundance of Positions**

There are very few people who have the required skill set to become a complete Data Scientist. This makes Data Science less saturated as compared with other IT sectors. Therefore, Data Science is a vastly abundant field and has a lot of opportunities. The field of Data Science is high in demand but low in supply of Data Scientists.

3. **A Highly Paid Career**

Data Science is one of the most highly paid jobs. According to Glassdoor, Data Scientists make an average of $116,100 per year. This makes Data Science a highly lucrative career option.

4. **Data Science is Versatile**

There are numerous applications of Data Science. It is widely used in healthcare, banking, consultancy services, and e-commerce industries. Data Science is a very versatile field. Therefore, you will have the opportunity to work in various fields.

5. **Data Science Makes Data Better**

Companies require skilled Data Scientists to process and analyze their data. They not only analyze the data but also improve its quality. Therefore, Data Science deals with enriching data and making it better for their company.

6. **Data Scientists are Highly Prestigious**

Data Scientists allow companies to make smarter business decisions. Companies rely on Data Scientists and use their expertise to provide better results to their clients. This gives Data Scientists an important position in the company.

7. **No More Boring Tasks**

Data Science has helped various industries to automate redundant tasks. Companies are using historical data to train machines in order to perform repetitive tasks. This has simplified the arduous jobs undertaken by humans before.

8. **Data Science Makes Products Smarter**

Data Science involves the usage of Machine Learning which has enabled industries to create better products tailored specifically for customer experiences. For example, Recommendation Systems used by e-commerce websites provide personalized insights to users based on their historical purchases. This has enabled computers to understand human-behaviour and make data-driven decisions.

9. **Data Science can Save Lives**

Healthcare sector has been greatly improved because of Data Science. With the advent of machine learning, it has been made easier to detect early-stage tumours. Also, many other health-care industries are using Data Science to help their clients.

10. **Data Science Can Make You A Better Person**

Data Science will not only give you a great career but will also help you in personal growth. You will be able to have a problem-solving attitude. Since many Data Science roles bridge IT and Management, you will be able to enjoy the best of both worlds.

Now that we are clear on why to choose data science, lets hop on to out next question:

How is it going to help me in my career?

Data Science has always been interesting to me. The way a dataset, which is clearly not possible for a normal human being (which has no connection to data science field), with the help of some analyzing tools, we can understand the data and hence can do the required task or action needed to be done.

After having the experience to analyze data, I am so interested in analyzing more and more datasets and hopefully land a job as a Machine Learning Engineer (which is part of Data science Team, which also requires knowledge on Machine learning algorithms as well as data science), and do something effective for the society.

At last I want to say, this was a very crucial step to the world of data science for me and I hope to keep learning in the future.

REFERENCES

Websites referred:

- **https://www.wikipedia.org (accessed on 4th September)**
- **https://www.upgrad.com (accessed on 4th September)**
- **https://www.discoverdatascience.org (accessed on 6th September)**
- **https://www.analyticsvidhya.com (accessed on 7th September)**
- **https://www.geeksforgeeks.org (accessed on 5th September)**
- **https://www.tutorialgateway.org (accessed on 9th September)**
- **https://prowebscraper.com (accessed on 9th September)**
- **https://www.ibm.com (accessed on 5th September)**
- **https://www.javapoint.com (accessed on 7th September)**
- **https://www.analyticsinsight.net (accessed on 10th September)**
- **https://www.educba.com (accessed on 29th August)**
- **https://www.google.com(accessed on 29thAugust, 4th,5th,6th,7th,9th,10th September)**
- **https://www.lifewire.com (accessed on 9th September)**
- **https://www.mode.com (accessed on 10th September)**

Book referred:

- Data Science and Big Data Analytics: Discovering, Analyzing, Visualizing and Presenting Data, 1st Edition, by EMC Education Services, Wiley, 5 January 2015