

**SRI VENKATESHWARA COLLEGE OF ENGINEERING AND TECHNOLOGY
(AUTONOMOUS)**

R.V.S Nagar, Chittoor – 517 127. (A.P)

(Approved by AICTE, New Delhi, Affiliated to JNTUA, Anantapur)

(Accredited by NBA, New Delhi & NAAC A+, Bangalore)

(An ISO 9001:2000 Certified Institution)

2023-2024



INTERNSHIP REPORT

A report submitted in partial fulfilment of the requirements for the Award
of Degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE & ENGINEERING

(DATA SCIENCE)

BY

GOTHAM ANUPA

Regd.No.21781A3240

Under supervision of

APSSDC

MRS.Hema,trainer,

(Duration: 17/04/2023 to 23/09/2023)

**SRI VENKATAESHWARA COLLEGE OF ENGINEERING AND TECHNOLOGY
(AUTONOMOUS)**

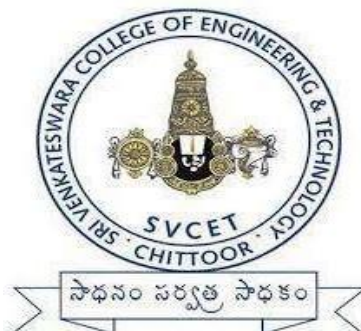
R.V.S Nagar, Chittoor – 517 127. (A.P)

(Approved by AICTE, New Delhi, Affiliated to JNTUA, Anantapur)

(Accredited by NBA, New Delhi & NAAC A+, Bangalore)

(An ISO 9001:2000 Certified Institution)

2023-2024



CERTIFICATE

This is to certify that the “Internship report” submitted by GOTHAM ANUPA (Regd.No.:21781A3240) is work done by him and submitted during 2023-2024.Academic year, in partial fulfilment of the requirements for the award of the Degree of **BACHELOR OF TECHNOLOGY in COMPUTER SCIENCE & ENGINEERING (DATA SCIENCE)**, at APSSDC.

MR.M. NAVALAN
Internship coordinator

Dr.M.LAVANYA
Head of the Department
(DATA SCIENCE)

CERTIFICATE

Roll No: 21781A3240

Certificate No: ACET23421



**Andhra Pradesh State Skill Development Corporation (APSSDC)
Indo-Euro Synchronization (IES) Pvt Ltd**

CERTIFICATE OF ACHIEVEMENT

This Certificate is Presented for Honorable Achievement to

Mr./Ms. Gotham Anupa,

Sri Venkateswara College of Engineering and Technology

For Successfully completing the Internship with **A Grade**

in “Certification on Emerging Technologies - **Data Science”**

During 17th April to 23rd September – 2023

Sessions conducted by Industry and International Partners

Certification & Knowledge Partners



**Indo-Euro
Synchronization Pvt Ltd**
Education and Research Resources

A. Hema

Mrs. A Hema

Project Head

Indo-Euro Synchronization Pvt Ltd

ACKNOWLEDGEMENT

- ❖ A Grateful thanks to **Dr.R.VENKATASWAMY** Chairman of Sri Venkateshwara College of Engineering & Technology(Autonomous) for providing education in their esteemed institution. I wish to record my deep sense of gratitude and profound thanks to our beloved Vice Chairman, Sri R.V. Srinivas for his valuable support throughout the course.
- ❖ I express our sincere thanks to **Dr.M.MOHAN BABU**, our beloved principal for his encouragement and suggestion during the course of study.
- ❖ With the deep sense of gratefulness, I acknowledge **Dr.M.LAVANYA** Head of the Department, Computer Science Engineering (CSD), for giving us inspiring guidance in undertaking internship.
- ❖ I express our sincere thanks to the internship coordinator **Mr.M.NAVALAN**, for his keen interest, stimulating guidance, constant encouragement with our work during all stages, to bring this report into fruition.
- ❖ I wish to convey my gratitude and sincere thanks to all members for their support and cooperation rendered for successful submission of report.
- ❖ Finally, I would like to express my sincere thanks to all teaching, non-teaching faculty members, our parents, and friends and for all those who have supported us to complete the internship successfully.

(NAME: GOTHAM ANUPA)
(ROLL.NO. 21781A3240)

ABSTRACT

Data science encompasses a set of principles, problem definitions, algorithms, and processes for extracting non obvious and useful patterns from large data sets. Many of the elements of data science have been developed in related fields such as machine learning and data mining. In fact, the terms data science, machine learning, and data mining are often used interchangeably. The commonality across these discipline focus on improving decision making through the analysis of data. However, although data science borrows from these other fields, it is broader in scope. Machine learning (ML) focuses of the design and evaluation of algorithms for extracting patterns from data. Data mining generally deals with the analysis of structured data and often implies an emphasis on commercial applications. Data science takes all of these considerations into account but also takes up other challenges, such as the capturing, cleaning, transforming of unstructured social media and web data; the use of big-data technologies to store and process big ,unstructured datasets; and questions related to data ethics and regulation.

ORGANIZATION PROFILE

APSSDC is a professionally managed company with years of industry experience in developing and delivering Enterprise specific Software and Web development solutions using latest technologies. Quality is the buzz word in today's world without which no organization can survive. Along with quality we at APSSDC. "Think Beyond" to take one step ahead and focus on Delivery of the solutions. We design processes that focus not just only on quality but also on delivery which increases the value to our global clients Apart from training our employees on latest technologies, we also empower them to deliver exciting solutions to our clients. At the core APSSDC operates in three specific domains namely Software Development, Website Design & Development and Geographic Information Services. We also offer our services in building E-Commerce solutions, Search Engine Optimization (SEO) and Database Administration services. Under each division we further provide specific industry solutions on focused domains with cutting edge technologies. We emphasize on building relationships with our clients by delivering projects on time and within budget.

Programs and opportunities

During the pandemic, the placement scenario took a hit. One major reason behind this that a lot of institutions were not able to host campus placements. Through this online recruitment fair, we aim to connect

all the aspiring freshers from across the nation with employers. The online fair offers a platform for the recruitment community to hire, and for the students community to apply for jobs and internships from the safety of their homes.

Methodologies:

There are many reasons why public sector customers are migrating to the cloud. Some are migrating to increase the productivity of their workforce. Others are looking to consolidate data centers or minimize costly infrastructure sprawl and modernize legacy applications that have lost value over time. Additionally, visionary organizations are re-imagining their missions by upgrading to cloud-hosted technologies that drive digital transformation.

INDEX

CHAPTER 1: INTRODUCTION TO DATA SCIENCE

- What is data science
- Importance of data science
- Applications of data science

CHAPTER 2 : SQL

- Introduction to sql
- Sql queries
- Relational data base concepts

CHAPTER 3:EXCEL

- Spread sheet
- Basic formulas and functions
- Import data into excel

CHAPTER 4:PYTHON PROGRAMMING

- Conditional statements
- String methods and functions
- Accessing elements and slicing

CHAPTER 5:NUMPY

- Introduction to numpy array
- Accessing elements
- Slicing

CHAPTER 6: TABLEAU

- Introduction to tableau
- Tableau charts
- Concepts related to tableau

CHAPTER 7:POWER BI

- Introduction to power bi service
- Components of power bi
- Applications of power bi

CHAPTER 8:FINAL PROJECT

CHAPTER 9:CONCLUSION

ABOUT TRAINING:

The Data Science Training by APSSDC is a 3 months online training program in which APSSDC aim to provide you with a comprehensive introduction to data science. In this training program, you will learn the basics of python, statistics, predictive modeling, excel, numpy, power bi ,tableau. This training program has video tutorials and is packed with assignments, assessments tests, quizzes, and practice exercises for you to get a hands-on learning experience.

Learning Objectives/Internship Objectives

Internships are generally thought of to be reserved for college students looking to gain experience in a particular field. However, a wide array of people can benefit from Training Internships in order to receive real world experience and develop their skills.

An objective for this position should emphasize the skills you already possess in the area and your interest in learning more

Internships are utilized in a number of different career fields, including architecture, engineering, healthcare, economics, advertising and many more.

Some internship is used to allow individuals to perform scientific research while others are specifically designed to allow people to gain first-hand experience working.

Utilizing internships is a great way to build your resume and develop skills that can be emphasized in your resume for future jobs. When you are applying for a Training Internship, make sure to highlight any special skills or talents that can make you stand apart from the rest of the applicants so that you have an improved chance of landing the position.

WEEKLY OVER VIEW OF INTERNSHIP ACTIVITIES

1st WEEK

	DAY	NAME OF THE MODULE/TOPICS COMPLETED
17-04-2023	Monday	Introduction to data science
20-04-2023	Thursday	Data science Overview
22-04-2023	Saturday	Importance of data science
25-04-2023	Tuesday	Applications of data science
29-04-2023	Saturday	Structured Query language
04-05-2023	Thursday	SQL Commands

2nd WEEK

	DAY	NAME OF THE MODULE/TOPICS COMPLETED
08-05-2023	Monday	SQL Queries
11-05-2023	Thursday	Modifying and analysing data with sql
15-05-2023	Monday	Relational Data base concepts
19-05-2023	Friday	Excel
23-05-2023	Tuesday	Excel basics for data analysis
27-05-2023	Saturday	Basic Formulas for excel

3rd WEEK

	DAY	NAME OF THE MODULE/TOPICS COMPLETED
31-05-2023	Wednesday	Functions of excel
05-06-2023	Monday	Import data into excel
09-06-2023	Friday	Introduction to python
14-06-2023	wednesday	Conditional Statements
19-06-2023	Monday	String methods
23-06-2023	Friday	String functions

4th WEEK

	DAY	NAME OF THE MODULE/TOPICS COMPLETED
27-06-2023	Tuesday	Accessing elements and slicing
01-07-2023	Saturday	(Module test)
05-07-2023	Wednesday	Introduction to numpy array
11-07-2023	Tuesday	Accessing the array elements
17-07-2023	Monday	Numpy Data types,Slicing
22-07-2023	Saturday	Sorting,Searching methods

5th WEEK

	DAY	NAME OF THE MODULE/TOPICS COMPLETED
25-07-2023	Tuesday	(Module test)
28-07-2023	Friday	Tableau
01-08-2023	Tuesday	Introducing tableau business science
05-08-2023	Saturday	Tableau charts
09-08-2023	Wednesday	Data science with tableau
14-08-2023	Monday	(Module test)

6th WEEK

	DAY	NAME OF THE MODULE/TOPICS COMPLETED
18-08-2023	Friday	Power Bi
22-08-2023	Tuesday	Introduction to power bi service
26-08-2023	Saturday	Microsoft power bi
28-08-2023	Monday	Architecture of power bi
30-08-2023	Wednesday	Components of power bi
01-09-2023	Friday	Features of power bi

7th WEEK

	DAY	NAME OF THE MODULE/TOPICS COMPLETED
03-09-2023	Sunday	Applications of power bi
05-09-2023	Tuesday	(Module test)
07-09-2023	Thursday	Industrial Expert talk
09-09-2023	Saturday	Awareness of Data science projects
12-09-2023	Tuesday	Orientation
14-09-2023	Thursday	(MODULE TEST)

8th WEEK

	DAY	NAME OF THE MODULE/TOPICS COMPLETED
16-09-2023	Saturday	Assessments
18-09-2023	Monday	Final Module Test
19-09-2023	Tuesday	Final project
20-09-2023	Wednesday	Working on Final Project
21-09-2023	Thursday	Submission of Final Project
23-09-2023	Saturday	Conclusion

INTRODUCTION TO DATA SCIENCE

Data Science Overview

Data science is the study of data. Like biological sciences is a study of biology, physical sciences, it's the study of physical reactions. Data is real, data has real properties, and we need to study them if we're going to work on them. Data Science involves data and some signs. It is a process, not an event. It is the process of using data to understand too many different things, to understand the world. Data Science is about data gathering, analysis and decision-making. Data Science is about finding patterns in data, through analysis, and make future predictions. By using Data Science, companies are able to make:

- Better decisions (should we choose A or B)
- Predictive analysis (what will happen next?)
- Pattern discoveries (find pattern, or maybe hidden information in the data)

By using Data Science, companies are able to make:

- Better decisions (should we choose A or B)
- Predictive analysis (what will happen next?)
- Pattern discoveries (find pattern, or maybe hidden information in the data)
- For route planning: To discover the best routes to ship
- To foresee delays for flight/ship/train etc. (through predictive analysis)
- To create promotional offers
- To find the best suited time to deliver goods
- To forecast the next years revenue for a company
- To analyze health benefit of training
- To predict who will win elections

Predictive modeling:

Predictive modeling is a form of artificial intelligence that uses data mining and probability to forecast or estimate more granular, specific outcomes. For example, predictive modeling could help identify customers who are likely to purchase our new One AI software over the next 90 days.

Machine Learning:

Machine learning is a branch of artificial intelligence (ai) where computers learn to act and adapt to new data without being programmed to do so. The computer is able to act independently of human interaction.

Forecasting:

Forecasting is a process of predicting or estimating future events based on past and present data and most commonly by analysis of trends. "Guessing" doesn't cut it.

Importance of Data Science

The importance of data Science brings together the domain expertise from programming, mathematics, and statistics to create insights and make sense of data. When we think about why data science is increasingly becoming important, the answer lies in the fact that the value of data is soaring heights. Did you know that Southwest Airlines, at one point was able to save \$100 million by leveraging data? They could reduce their planes' idle time that waited at the tarmac and drive a change in utilizing their resources. In short, today, it is not possible for any business to imagine a world without data.

1. It helps brands to understand their customers in a much enhanced and empowered manner
Customers are the soul and base of any brand and have a great role in their success and failure. With data science, brands can connect with their customers in a personalized manner, thereby ensuring better brand power and engagement.
2. One of the reasons why it is gaining so much attention is because it allows brands to communicate their story in such an engaging and powerful manner. When brands and companies comprehensively utilize this data, they can share their story with their target audience, creating better brand connections. After all, nothing connects with consumers like an effective and powerful story that can inculcate all human emotions.
3. Big Data is a new field that is constantly growing and evolving. With so many tools being developed, big data is almost regularly helping brands and organizations solve complex problems in IT, human resources, and resource management effectively and strategically. This means effective use of resources, both material and non-material.

The Vital Importance of Data Science in Today's Data-Driven World



Data science can add value to any business who can use their data well. From statistics and insights across workflows and hiring new candidates, to helping senior staff make better-informed decisions, data science is valuable to any company in any industry.

Interested in a career in Data Science? Simpli learn offers the perfect Data Science course to help you learn everything you need to know about getting started as a Data Scientist.

Applications of Data Science:

Data science and big data are making an undeniable impact on businesses, changing day-to-day operations, financial analytics, and especially interactions with customers. It's clear that businesses can gain enormous value from the insights data science can provide. But sometimes it's hard to see exactly how. So let's look at some examples. In this era of big data, almost everyone generates masses of data every day, often without being aware of it. This digital trace reveals the patterns of our online lives.

Healthcare: Data science can identify and predict disease, and personalize healthcare recommendations.

Transportation: Data science can optimize shipping routes in real-time.

Sports: Data science can accurately evaluate athletes' performance.

Government: Data science can prevent tax evasion and predict incarceration rates.

E-commerce: Data science can automate digital ad placement.

Gaming: Data science can improve online gaming experiences.

Social media: Data science can create algorithms to pinpoint compatible partners.

Fintech: Data science can help create credit reports and financial profiles, run accelerated underwriting and create predictive models based on historical payroll data.

APPLICATIONS OF DATA SCIENCE



CHAPTER-2:SQL

Introduction to sql

SQL: It is a standard language for accessing and manipulating databases.

What is SQL?

- SQL stands for Structured Query Language.
- SQL lets you access and manipulate databases.
- SQL became a standard of the American National Standards Institute (ANSI) in 1986, and of the International Organization for Standardization (ISO) in 1987.

What Can SQL do?

- SQL can execute queries against a database
- SQL can retrieve data from a database
- SQL can insert records in a database
- SQL can update records in a database
- SQL can delete records from a database
- SQL can create new databases
- SQL can create new tables in a database
- SQL can create stored procedures in a database
- SQL can create views in a database
- SQL can set permissions on tables, procedures, and views

RDBMS

- RDBMS stands for Relational Database Management System.
- RDBMS is the basis for SQL, and for all modern database systems such as MS SQL Server, IBM DB2, Oracle, MySQL, and Microsoft Access.
- The data in RDBMS is stored in database objects called tables. A table is a collection of related data entries and it consists of columns and rows.

Example:

Sql statement:

```
Select *from customers;
```

Every table is broken up into smaller entities called fields. The fields in the Customers table consist of Customer ID, Customer Name, Contact Name, Address, City, PostalCode and Country. A field is a column in a table that is designed to maintain specific information about every record in the table.

A record, also called a row, is each individual entry that exists in a table. For example, there are 91 records in the above Customers table. A record is a horizontal entity in a table.

A column is a vertical entity in a table that contains all information associated with a specific field in a table.

Sql statements:

- Most of the actions you need to perform on a database are done with SQL statements.
- SQL statements consists of keywords that are easy to understand.
- The following SQL statement returns all records from a table named "Customers":

SQL QUERIES

- **SELECT** - extracts data from a database
- **UPDATE** - updates data in a database
- **DELETE** - deletes data from a database
- **INSERT INTO** - inserts new data into a database
- **CREATE DATABASE** - creates a new database
- **ALTER DATABASE** - modifies a database
- **CREATE TABLE** - creates a new table
- **ALTER TABLE** - modifies a table
- **DROP TABLE** - deletes a table
- **CREATE INDEX** - creates an index (search key)
- **DROP INDEX** - deletes an index

The SQL INSERT INTO Statement

The SQL **INSERT INTO Statement** is used to add new rows of data into a table in the database.

Almost all the RDBMS provide this SQL query to add the records in database tables.

Syntax:

```
INSERT INTO table_name(col1 ,col2,..)VALUES('v1','v2',...)
```

Example:

```
INSERT INTO Customers (Customer Name, City, Country)  
VALUES ('Cardinal', 'Stavanger', 'Norway');
```

The SQL SELECT Statement

The SQL **SELECT Statement** is used to fetch the data from a database table which returns this data in the form of a table. These tables are called result-sets.

Syntax

There are two basic syntaxes of the SQL **INSERT INTO** statement which are shown below :

Select col 1, col 2, from table name;

The basic syntax of the SELECT Query is as follows –

Example:

To see an example, let us create a table with name **CUSTOMERS** in the MySQL database using the CREATE TABLE statement as shown below –

```
SELECT Customer Name, City FROM Customers;
```

The SQL UPDATE Statement

The SQL **UPDATE** Statement is used to modify the existing records in a table. This statement is a part of Data Manipulation Language (DML), as it only modifies the data present in a table without affecting the table's structure

Syntax

The basic syntax of the SQL UPDATE statement with a WHERE clause is as follows –

UPDATE table name

SET col =val1,val 2,....

WHERE condition;

Example

Assume we have created a table named CUSTOMERS using the CREATE TABLE statement as shown below


```
UPDATE Customers  
SET ContactName='Juan'  
WHERE Country='Mexico';
```

The SQL DELETE Statement

The SQL **DELETE** Statement is used to delete the records from an existing table. In order to filter the records to be deleted (or, delete particular records), we need to use the **WHERE** clause along with the DELETE statement.

Syntax

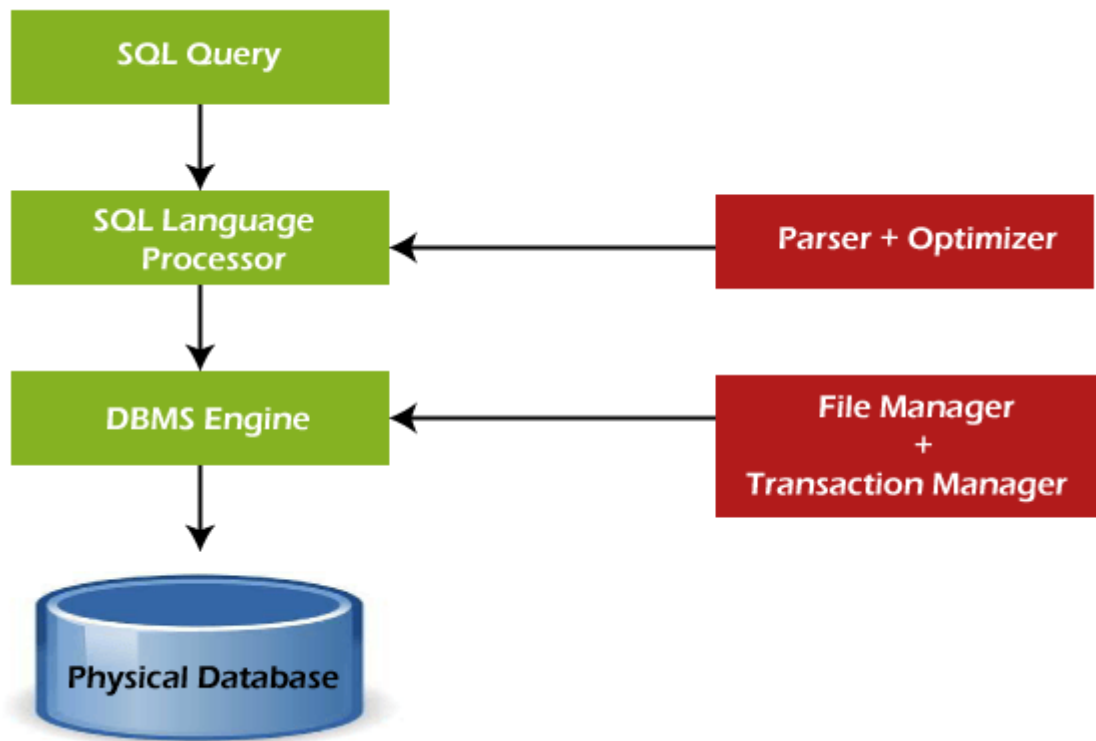
The basic syntax of the SQL DELETE Query with the WHERE clause is as follows –

```
DELETE from table_name;
```

Example

```
DELETE from customers;
```

Assume we have created a table named **CUSTOMERS** which contains the personal details of customers including their name, age, address and salary etc. as shown below –



Relational database concepts

SQL Constraints

Constraints are the rules enforced on data columns on a table. These are used to limit the type of data that can go into a table. This ensures the accuracy and reliability of the data in the database.

Constraints can either be column level or table level. Column level constraints are applied only to one column whereas, table level constraints are applied to the entire table.

1 NOT NULL Constraint

Ensures that a column cannot have a NULL value.

2 DEFAULT Constraint

Provides a default value for a column when none is specified.

3 UNIQUE Key

Ensures that all the values in a column are different.

- 4 **PRIMARY Key**
Uniquely identifies each row/record in a database table.
- 5 **FOREIGN Key**
Uniquely identifies a row/record in any another database table.
- 6 **CHECK Constraint**
Ensures that all values in a column satisfy certain conditions.
- INDEX Constraint**
Used to create and retrieve data from the database very quickly.

Data Integrity

The following categories of data integrity exist with each RDBMS –

- **Entity Integrity** – This ensures that there are no duplicate rows in a table.
- **Domain Integrity** – Enforces valid entries for a given column by restricting the type, the format or the range of values.
- **Referential integrity** – Rows cannot be deleted, which are used by other records.
- **User-Defined Integrity** – Enforces some specific business rules that do not fall into entity, domain or referential integrity.

Database Normalization

Database normalization is the process of efficiently organizing data in a database. There are two reasons of this normalization process –

- Eliminating redundant data, for example, storing the same data in more than one table.
- Ensuring data dependencies make sense.
- It is your choice to take it further and go to the Fourth Normal Form, Fifth Normal Form and so on, but in general, the Third Normal Form is more than enough for a normal Database Application.
- First Normal Form (1NF)
- Second Normal Form (2NF)
- Third Normal Form (3NF)

EXCEL

SPREAD SHEET

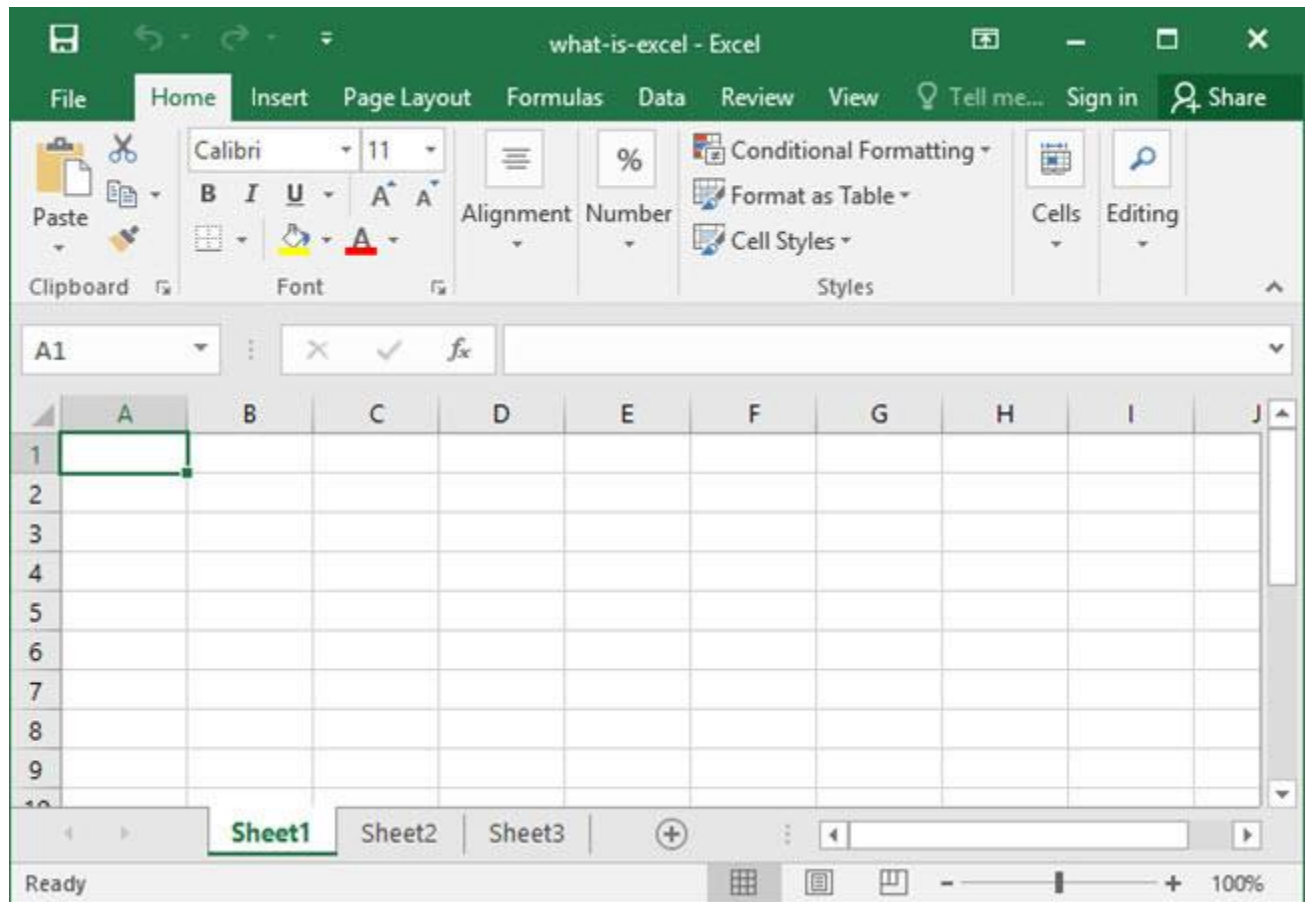
The following are just a few of the features available in most spreadsheet programs.

Cell formatting

Within the spreadsheet, selected cells can be formatted to represent various numeric values. For example, financial data can be given accounting formatting, which will apply decimal places and commas to represent dollars and cents.

Formulas

Under the formula bar, users can perform calculations on the contents of a cell against the contents of another cell. For example, if a person were using the spreadsheet to reconcile transactions they could highlight all the cells that need to be added up and insert a sum function.



Common spreadsheet applications

Daniel Bricklin and Bob Frankston created the first spreadsheet application, named VisiCalc for "visible calculator." It was popular on the Apple II, one of the first computers used by businesses.

Lotus 1-2-3 surpassed VisiCalc to become the program that cemented the IBM PC as the preeminent personal computer in business during the 1980s and 1990s. IBM acquired Lotus in 1995 and continued selling Lotus 1-2-3 through 2013, when it discontinued the spreadsheet application, which had fallen behind Microsoft Excel in the '90s and never recovered.

What are spreadsheet controls?

Spreadsheet controls are measures a business's accounting team uses to safeguard the integrity and accuracy of its bookkeeping practices and of financial records. It is an ongoing effort to rapidly detect and resolve errors and maintain the security of all data. Thorough training is important to the success of spreadsheet control programs.

Features of a spreadsheet control program should include:

- Access control (usernames, passwords, biometrics)
- Up-to-date list of authorized users
- Access restrictions on cells performing critical computations
- Strong encryption
- Outgoing data accuracy checks
- Incoming data accuracy checks
- Routine maintenance of network hardware
- Prevention of data loss/corruption through cross-referencing check sum processes
- Comprehensive testing of latest spreadsheet functions/features
- Scheduled archiving/incremental backup
- Storage provisions offsite
- Backup/archiving redundancy, use of multiple media types
- Data recovery procedures in place
- Ease of use and flexibility

Basic formulas and functions

1. SUM

The SUM() formula performs addition on selected cells. It works on cells containing numerical values and requires two or more cells.

Ex:= sum(c2:c5)

2. MIN and MAX

The MIN() formula requires a range of cells, and it returns the minimum value. For example, we want to display the minimum weight among all athletes on the E6 cell. The MIN formula will search for the minimum value and show 60.

Ex:=Min(E2:E5)

The MAX() formula is the opposite of MIN(). It will return the maximum value from the selected range of cells. The formula will look for the maximum value and return 82.

Ex:=Max(E2:E5)

3. AVERAGE

The AVERAGE() formula calculates the average of selected cells. You can provide a range of cells (C2:C5) or select individual cells (C2, C3, C5).

Ex:=Average(C2:C5)

4. COUNT

The COUNT() formula counts the total number of selected cells. It will not count the blank cells and different data formats other than numeric.

Ex:=count(E2:E5)

5. POWER

In the beginning, we learned to add power using “^”, which is not an efficient way of applying power to a cell.

Instead, we recommended using the POWER() formula to square, cube, or apply any raise to power to your cell.

Ex:=power(D2/100,2)

6. CEILING and FLOOR

The CEILING() formula rounds a number **up** to the nearest given multiple. In our case, we will round 3.24 up to a multiple of 1 and get 4. If the multiple is 5, it will round up the number 3.24 to 5.

Ex:=CEILING(F2,1)

The FLOOR() rounds a number **down** to the nearest given multiple. As we can see in the image below, instead of converting 3.24 to 4, it has rounded the number to 3.

EX:FLOOR(F2,1)

7. CONCAT

The CONCAT() Excel formula joins or merges multiple strings or cells with strings into one. For example, if we want to join the age and sex of the athletes, we will use CONCAT. The formula will automatically convert a numeric value from age to string and combine it.

EX:“24”+“M” = “24M”

8. TRIM

TRIM is used to remove extra spaces from the start, middle, and end. It is commonly used to identify duplicate values in cells, and for some reason, extra space makes it unique.

For example:=TRIM(A4)a

IMPORT DATA INTO EXCEL

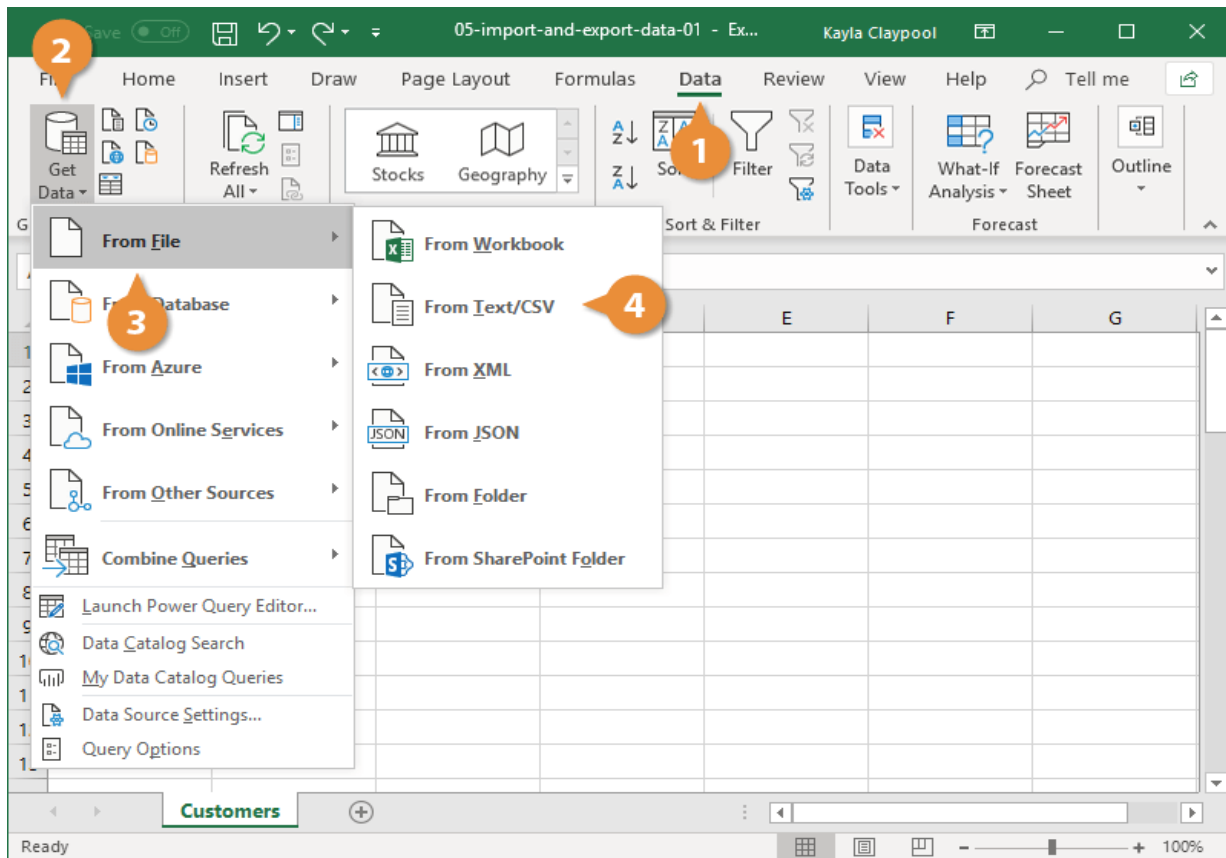
Import Data

Excel can import data from external data sources including other files, databases, or web pages.

1. Click the **Data** tab on the Ribbon..
2. Click the **Get Data** button.

Some data sources may require special security access, and the connection process can often be very complex. Enlist the help of your organization's technical support staff for assistance.

3. Select **From File**.
4. Select **From Text/CSV**.



5. Select the file you want to import.

6. Click **Import**.

PYTHON PROGRAMMING

Conditional Statements In Python

In programming languages, most of the time in large projects we have to control the flow of execution of our program and we want to execute some set of statements only if the given condition is satisfied, and a different set of statements when it's not satisfied.

Conditional statements are also known as decision-making statements. We need to use these conditional statements to execute the specific block of code if the given condition is true or false.

In Python we can achieve decision making by using the following statements:

- if statements
- if-else statements
- elif statements
- Nested if and if-else statements
- elif ladder

#1) if statements

Python if statement is one of the most commonly used conditional statements in programming languages.

It decides whether certain statements need to be executed or not. It checks for a given condition, if the condition is true, then the set of code present inside the "if" block will be executed otherwise not.

The if condition evaluates a Boolean expression and executes the block of code only when the Boolean expression becomes TRUE.

SYNTAX:

If (`EXPRESSION == TRUE`):

Block of code

else:

Block of code

EXAMPLE 1:

```
num = 5
if (num < 10):
    print("Num is smaller than 10")

print("This statement will always be executed")
```

#2) if-else statements

The statement itself says if a given condition is true then execute the statements present inside the “if block” and if the condition is false then execute the “else” block.

The “else” block will execute only when the condition becomes false. It is the block where you will perform some actions when the condition is not true.

SYNTAX

```
If (EXPRESSION == TRUE):
    Statement (Body of the block)
else:
    Statement (Body of the block)
```

EXAMPLE 1:

```
num = 5
if(num > 10):
    print("number is greater than 10")
else:
    print("number is less than 10")

print("This statement will always be executed")
```

#3) elif statements

In Python, we have one more conditional statement called “elif” statements. “elif statement is used to check multiple conditions only if the given condition is false.

It’s similar to an “if-else” statement and the only difference is that in “else” we will not check the condition but in “elif” we will check the condition.”elif” statements are similar to “if-else” statements but “elif” statements evaluate multiple conditions.

Syntax:

if (condition):

elif (condition):

else:

#Set of statement to be executed when both if and elif conditions are false

Example: 1

num = 10

if (num == 0):

print("Number is Zero")

elif (num > 5):

print("Number is greater than 5")

else:

print("Number is smaller than 5")

#4) Nested if-else statements

Nested “if-else” statements mean that an “if” statement or “if-else” statement is present inside another if or if-else block. Python provides this feature as well, this in turn will help us to check multiple conditions in a given program. An “if” statement is present inside another “if” statement which is present inside another “if” statements and so on.

SYNTAX

if(condition):

 #Statements to execute if condition is true

if(condition):

 #Statements to execute if condition is true

 #end of nested if

#end of if

EXAMPLE 1:

num = 5

if(num >0):

print("number is positive")

if(num<10):

print("number is less than 10")

5) *elif Ladder*

We have seen about the “elif” statements but what is this elif ladder? As the name itself suggests a program that contains a ladder of “elif” statements or “elif” statements are structured in the form of a ladder.

This statement is used to test multiple expressions.

Syntax:

if (condition):

elif (condition):

elif (condition):

elif (condition):

else:

all if and elif conditions are false

Example: 1

My marks = 90

if (my marks < 35):

print("Sorry!, You failed the exam")

elif(my marks > 60 **and** my marks > 100):

print("Passed **in** First **class**")

else:

print("Passed **in** First **class** with distinction")

STRING METHODS AND FUNCTIONS

Python String Methods

Python has 47 methods for handling strings. There are almost as many built-in Python functions as there are string methods. Which string operations ought you learn first?

There are about a dozen string techniques that are quite beneficial and should be memorized.

We'll quickly review the other approaches and the reasons they're less successful after looking at the most beneficial string ways.

Table of Python String Methods

Function Name	Description
capitalize()	the string's initial character is changed to an uppercase (capital) letter.
casefold()	uses case-free string matching
center()	Add the requested character to the string as padding.
count()	The number of times a substring appears in the string is returned.
encode()	Strings are encoded using the chosen encoding technique.
endswith()	If a string ends with the specified suffix, it returns "True."

expandtabs()	specifies how much space will be used to replace the string's "t" symbol.
find()	if the substring is discovered, returns the lowest index of the substring.
format()	prepares the string for console output.
format_map()	uses a dictionary to format given values in a string.
index()	returns the location of a substring's first occurrence within a string.

Python String Functions

There are several ways to manipulate a String. It's impossible to recall them all, though.

So I've separated them into several categories here.

Must Know String Functions

Function	Description
format()	It's used to create a formatted string from the template string and the supplied values.

split()	Python string split() function is used to split a string into the list of strings based on a delimiter.
join()	This function returns a new string that is the concatenation of the strings in iterable with string object as a delimiter.
strip()	Used to trim whitespaces from the string object.
format_map()	Python string format_map() function returns a formatted version of the string using substitutions from the mapping provided.
upper()	We can convert a string to uppercase in Python using str.upper() function.
lower()	This function creates a new string in lowercase.
replace()	Python string replace() function is used to create a new string by replacing some parts of another string.

ACCESSING ELEMENTS AND SLICING

What is accessing elements in Python?

The syntax for accessing the elements of a list is the same as the syntax for accessing the characters of

a string.

What is slicing in Python?

Slicing is the extraction of a part of a string, list, or tuple. It enables users to access the specific range of elements by mentioning their indices. Syntax: Object [start:stop:step] “Start” specifies the starting index of a slice. “Stop” specifies the ending element of a slice.

The `slice()` function returns a slice object.

A slice object is used to specify how to slice a sequence. You can specify where to start the slicing, and where to end. You can also specify the step, which allows you to e.g. slice only every other item.

Syntax

```
slice(start, end, step)
```

EXAMPLE:

```
a = ("a", "b", "c", "d", "e", "f", "g", "h")
```

```
x = slice(3, 5)
```

```
print(a[x])
```

Indexing

Indexing means referring to an element of an iterable by its position within the iterable. Each of a string's characters corresponds to an index number and each character can be accessed using its index number. We can access characters in a String in Two ways :

1. Accessing Characters by Positive Index Number
2. Accessing Characters by Negative Index Number

Accessing Characters by Positive Index Number: In this type of Indexing, we pass a Positive Index (which we want to access) in square brackets. The index number starts from index number 0 (which denotes the first character of a string).

G	e	e	k	s		f	o	r		G	e	e	k	s	!
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

G	e	e	k	s		f	o	r		G	e	e	k	s	!
-16	-15	-14	-13	-12	-11	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1

Example 1 (Positive Indexing) :

```
a='whatsapp'
```

```
print(a[0])      #'w'
```

```
print(a[3])      #'t'
```

Accessing Characters by Negative Index Number: In this type of Indexing, we pass the Negative index(which we want to access) in square brackets. Here the index number starts from index number -1 (which denotes the last character of a string).

Example 2 (Negative Indexing) :

```
Sr="happy birthday"
```

```
Print(Sr[-1])    #'y'
```

```
Print(Sr[-4])    #'h'
```

NUMPY

Importance of Numpy:

What is NumPy?

- NumPy is a Python library used for working with arrays.
- It also has functions for working in domain of linear algebra, fourier transform, and matrices.
- NumPy was created in 2005 by Travis Oliphant. It is an open source project and you can use it freely.
- NumPy stands for Numerical Python.

Why Use NumPy?

- In Python we have lists that serve the purpose of arrays, but they are slow to process.
- NumPy aims to provide an array object that is up to 50x faster than traditional Python lists.
- The array object in NumPy is called **ndarray**, it provides a lot of supporting functions that make working with **ndarray** very easy.
- Arrays are very frequently used in data science, where speed and resources are very important.

Installation of NumPy

If you have [Python](#) and [PIP](#) already installed on a system, then installation of NumPy is very easy.

Install it using this command:

```
C:\users:\your name>pip install numpy
```

If this command fails, then use a python distribution that already has NumPy installed like, Anaconda, Spyder etc.

Import NumPy

Once NumPy is installed, import it in your applications by adding the **import** keyword:

```
import numpy
```

```
ex: import numpy
```

```
    arr = numpy.array([1, 2, 3, 4,5])
```

```
    print(arr)                                # 1,2,3,4,5
```

ACCESSING ARRAY ELEMENTS

Indexing means referring to an element of an iterable by its position within the iterable. Each of a string's characters corresponds to an index number and each character can be accessed using its index number. We can access characters in a String in Two ways :

3. Accessing Characters by Positive Index Number
4. Accessing Characters by Negative Index Number

Accessing Characters by Positive Index Number: In this type of Indexing, we pass a Positive index(which we want to access) in square brackets. The index number starts from index number 0 (which denotes the first character of a string).

Ex:

```
mylist = ['apple', 'banana', 'cherry', 'date']  
print(mylist[0]) # output: 'apple'  
print(mylist[1]) # output: 'banana'
```

Accessing Characters by Negative Index Number: In this type of Indexing, we pass the Negative index(which we want to access) in square brackets. Here the index number starts from index number -1 (which denotes the last character of a string).

Ex:

```
mylist = ['apple', 'banana', 'cherry', 'date']  
print(mylist[-1]) # output: 'apple'  
print(mylist[-2]) # output: 'banana'
```

SLICING

What is slicing in Python?

Slicing is the extraction of a part of a string, list, or tuple. It enables users to access the specific range of elements by mentioning their indices. Syntax: Object [start:stop:step] “Start” specifies the starting index of a slice. “Stop” specifies the ending element of a slice.

The `slice()` function returns a slice object.

A slice object is used to specify how to slice a sequence. You can specify where to start the slicing, and where to end. You can also specify the step, which allows you to e.g. slice only every other item.

Syntax

`slice(start, end, step)`

EXAMPLE:

```
1.a = ("a", "b", "c", "d", "e", "f", "g", "h")
```

```
x = slice(3, 5)
```

```
print(a[x])           #('d', 'e')
```

```
2.a=("a", "b", "c", "d", "e", "f", "g", "h")
```

```
x= slice(0, 8, 3)
```

```
print(a[x])           #('a', 'd', 'g')
```

TABLEAU

INTRODUCTION TO TABLEAU:

Tableau was founded in 2003 as a result of a computer science project at Stanford that aimed to improve the flow of analysis and make data more accessible to people through visualization.

Co-founders Chris Stolte, Pat Hanrahan, and Christian Chabot developed and patented Tableau's foundational technology, VizQL—which visually expresses data by translating drag-and-drop actions into data queries through an intuitive interface. Since our foundation, we've continuously invested in research and development at an unrivaled pace, developing solutions to help anyone working with data to get to answers faster and uncover unanticipated insights.

This includes making machine learning, statistics, natural language, and smart data prep more useful to augment human creativity in analysis. And we not only offer a complete, integrated analytics platform, but also proven enablement resources to help customers deploy and scale a data-driven culture that drives resilience and value through powerful outcomes.

Tableau was acquired by [Salesforce](#) in 2019, and our mission remains the same: to help people see and understand their data. Today, organizations everywhere—from non-profits to global enterprises, and across all industries and departments—are empowering their people with Tableau to drive change with data.

"Tableau is seen as a business change program, to change the culture and way of thinking about data and decision-making. As we worked through our journey and showed people how easy it was to get insights themselves, the real eureka moments came out. Now we can take a step back and support them in their own journeys rather than having to do everything centrally."

GOVERNANCE IN TABLEAU

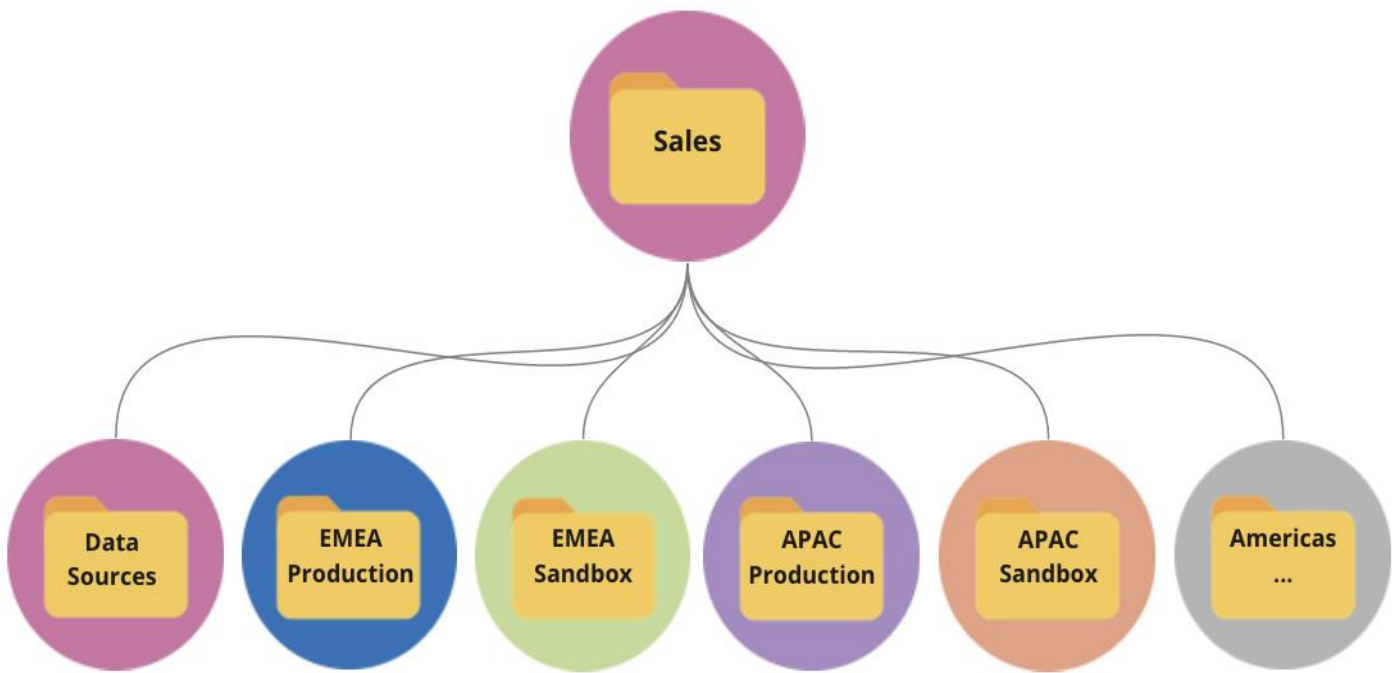
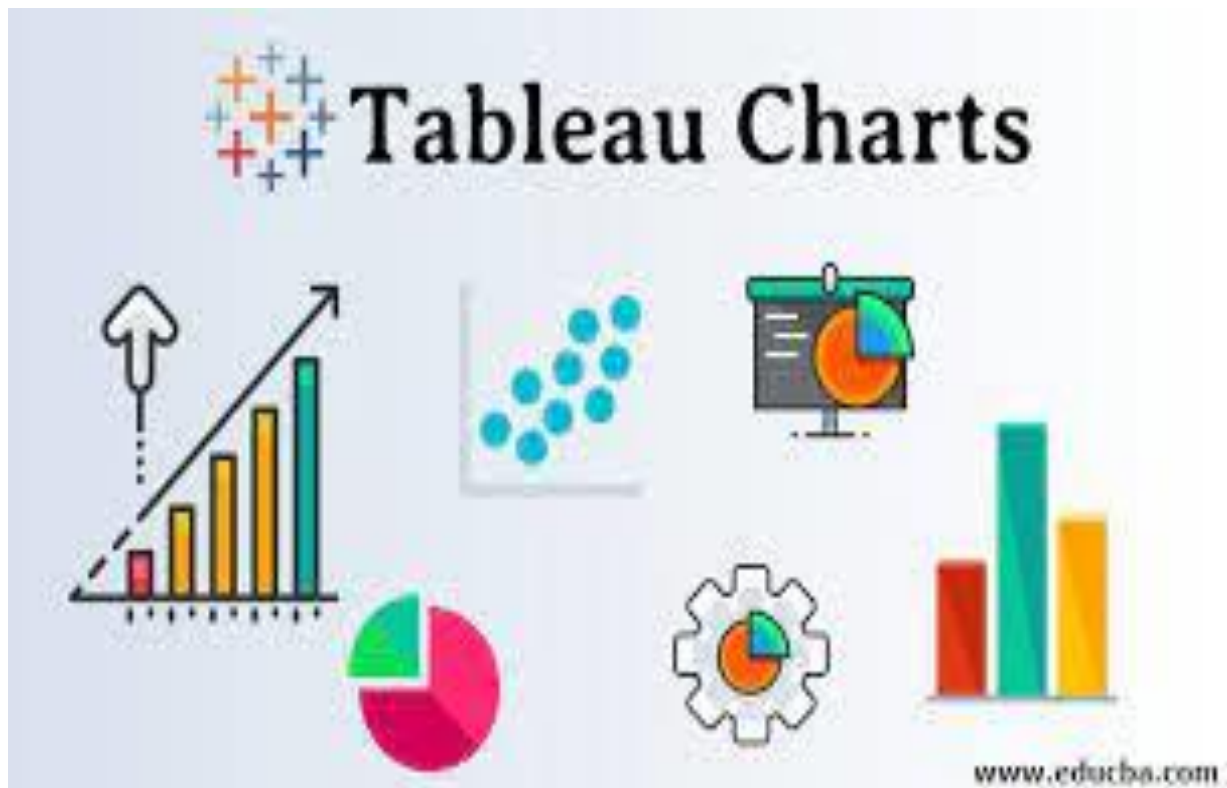


Tableau is a business intelligence, data analytics and data infrastructure platform, created to simplify data-driven decision-making. It was designed with a clear [mission](#), to "help people see and understand data."

TABLEAU CHARTS

Tableau is a data visualization tool that allows drawing different charts and graphs using Show me with just simple drag and drop of features.

Heat Map, Highlighted Table, Symbol Map, Filled Map, Pie Chart, Horizontal Bar Chart, Stacked Bar Chart, Side-by-Side Bar Chart, Tree Map, Circle View, Side-by-side Circle View, Line Chart (Continuous), Line Chart Discrete, Dual Line Chart, Area Chart (Continuous), Area Chart (Discrete), Dual Combination, Scatter Plot, Histogram, Box and Whisker Plot, Gantt Chart, Bullet Graph, Packed Bubble that are available depending on the number of dimensions, measure, bins, etc. of dimensions, measures, bins, etc



Bar Chart

- Represent data in rectangular bar
- Used to compare data across categories, highlight trends, differences, and outliers
- More effective when data can be split into multiple categories

Line Chart

- Line chart connects individual numeric data points
- Use to compare data over different periods
- A straightforward way to visualize change in one value relative to another
- Minimum Requirement: 1 Date, 0 or more dimensions, one or more measures

Pie Chart

- Represents Segment-wise data
- Used to show relative portion/percentage of information
- Powerful for adding detail to other visualization
- Distinct colors are used to describe different portions of features

Map

- Use to show geocoded data like postal code, state, country
- Highlights geographical trends in the most accessible and efficient way

Scatter Plot

- Used to visualize the relationship between two measures
- Scatter Plot investigates the relationship between different variables
- The plot is created when both row and column shelf have at least one measure

Gantt Plot

- Used for comparison of data between categories
- Use for displaying a project schedule
- It has time-series data that is available in a natural and easy to understand format

Bubble Chart

- Use to visualize measure and dimension in bubble form
- It shows the concentration of data along the axes
- Having different sizes and colors, it becomes easy to analyze

Histogram Chart

- A histogram displays the shape of the distribution
- Represents how data is distributed across different groups

- Used to understand the distribution of the data

Bullet Chart

- It is an indicator to show the performance of the measure
- Use to evaluate the performance of a matrix against a goal

Heat Map

- The best way to compare data across different categories is by using colors
- It shows the relationship between two features
- Created using one or more dimensions and a measure

Tree Map

- Rectangular chart representing data in nested rectangle
- Use to show hierarchical data as a portion of a whole
- It makes efficient use of space to display the entire data at once

Box-and-whisker Plot

- Box-and-Whisker plots are divided into two parts

Basic concepts of tableau

DATA VISUALIZATION

Data visualization is the graphical representation of information and data. Data visualization uses statistical graphics, plots, information graphics, and other tools to communicate information clearly and efficiently. Effective visualization helps users analyze and reason about data and evidence. It makes complex data more accessible, understandable, and usable.

Tableau Software is an interactive data visualization software company focused on business intelligence. **Tableau** is one of the data visualization tools which uses visual elements like charts, graphs, and maps, and provides an accessible way to see and understand trends, outliers, and patterns in data. In the commercial environment, data visualization is often referred to as **dashboards**. Info graphics are another very common form of data visualization.

A **dashboard** is a presentation of any number of related views and other elements (such as text or images) arranged together as a cohesive whole to communicate a message to an audience. Dashboards are often designed to be interactive.

In this article, we will have an overview of basic concepts and terminologies associated with **Tableau Desktop**. The purpose is to cover limited and most important topics, not the exhaustive topics of Tableau, in a short span of time.

Tableau Query Language

The unique and exciting experience of working with data in Tableau is a result of (**VizQL, Visual Query Language**).

Based on the requirement, it can then be translated into flavors of SQL (Structured Query Language), MDX (Multidimensional Expressions), and TQL (Tableau Query Language, used for extracted data).

Tableau automatically performs the translation of VizQL into a native query to be run by the source data engine. The web data connector allows you to write a connector for any online data you wish to retrieve. Additionally, for any database without a native connection, Tableau gives you the ability to use a generic ODBC connection.

The Extract API allows you to programmatically extract and combine any data sources for use in Tableau.

Measure & Dimension

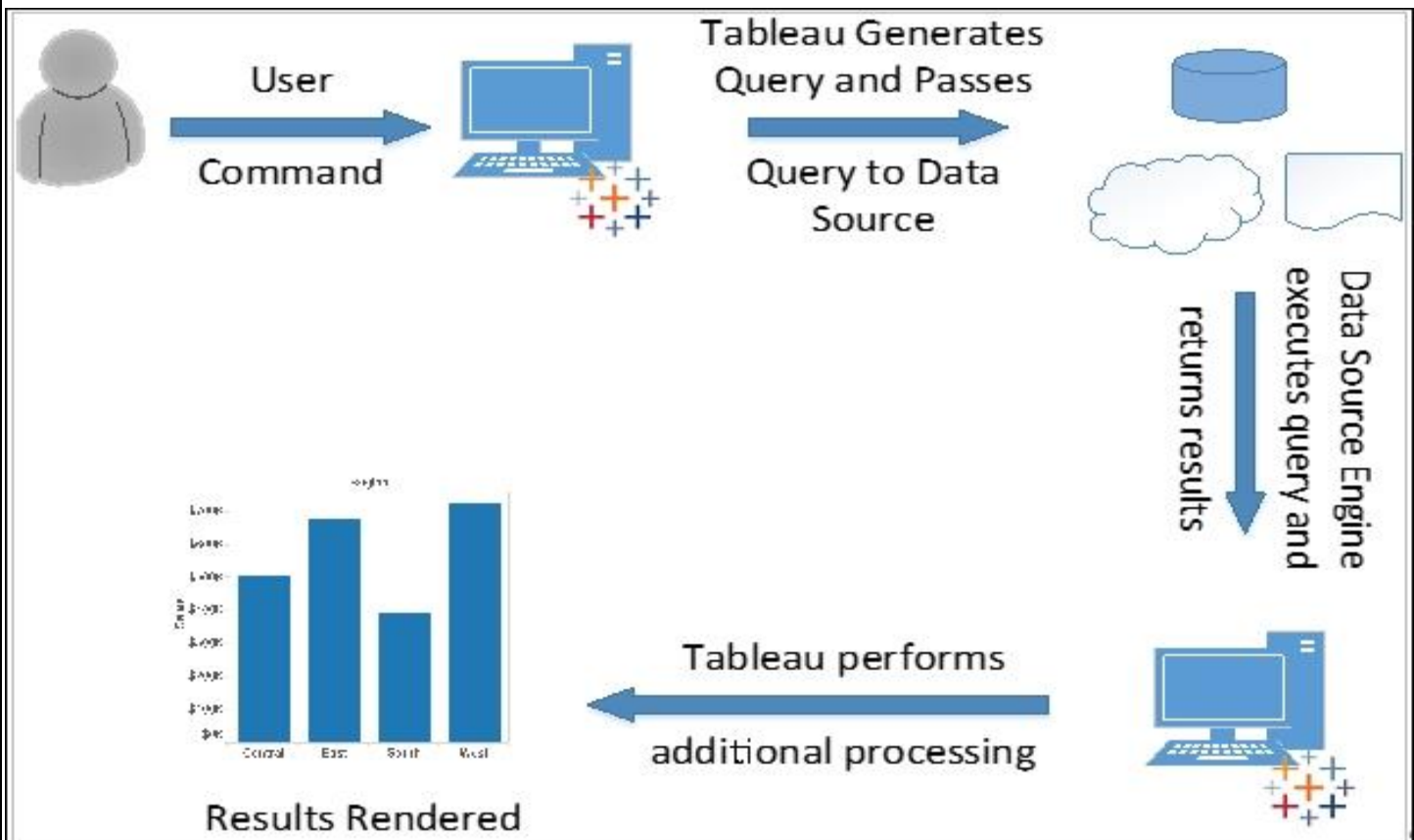
- Measures are values that are **aggregated**. For example, they are summed, averaged, counted, or have a minimum or a maximum.
- Dimensions are values that **determine the level of detail** at which measures are aggregated. The combination of dimensions used in the view defines the view's basic level of detail.

Live & Extract data connection

Live connections offer the convenience of **real-time updates**, with any changes in the data source reflected in Tableau. But live connections also rely on the database for all queries. With live connections, your data queries are only as fast as the database itself.

Tableau Data Extracts (.tde or .hyper file) are snapshots of data optimized for aggregation and **loaded into system memory** to be quickly recalled for visualization. Extracts tend to be much faster than live connections, especially in more complex visualizations with large data sets, filters, calculations, etc.

You may choose to use extracts to increase performance over traditional databases as live connection databases are not always optimized for fast performance.



POWER BI

Introduction to power bi service:

Microsoft Power BI is an interactive data visualization software product developed by Microsoft with a primary focus on business intelligence. It is part of the Microsoft Power Platform. Power BI is a collection of software services, apps, and connectors that work together to turn various sources of data into static and interactive data visualizations. Data may be input by reading directly from a database, structured files such as spreadsheets, CSV, XML, JSON, XLSX, and SharePoint.

Power BI provides cloud-based BI (business intelligence) services, known as "Power BI Services", along with a desktop-based interface, called "Power BI Desktop". It provides data warehouse capabilities including data preparation, data mining, and interactive dashboards. In March 2016, Microsoft released an additional service called Power BI Embedded on its Azure cloud platform. One main differentiator of the product is the ability to load custom visualizations.

Key components of the Power BI ecosystem are as follows:

Power BI Desktop

The Windows desktop-based application for PCs, primarily for designing and publishing reports to the service.

Power BI Service

The SaaS-based (software as a service) online service. This was formerly known as Power BI for Office 365, now referred to as PowerBI.com, or simply Power BI.

Power BI Mobile Apps

The Power BI Mobile apps for Android and iOS devices, as well as for Windows phones and tablets.

Power BI Gateway

Gateways are used to sync external data in and out of Power BI and are required for automated refreshes. In enterprise mode, it can also be used by Microsoft Power Automate (previously called Flows) and PowerApps in Office 365.

Power BI Embedded

Power BI REST API can be used to build dashboards and reports into the custom applications that serves Power BI users, as well as non-Power BI users.

Power BI Report Server

An on-premises Power BI is a reporting product for companies that choose not to store data in the cloud-based Power BI Service.

Power BI Premium

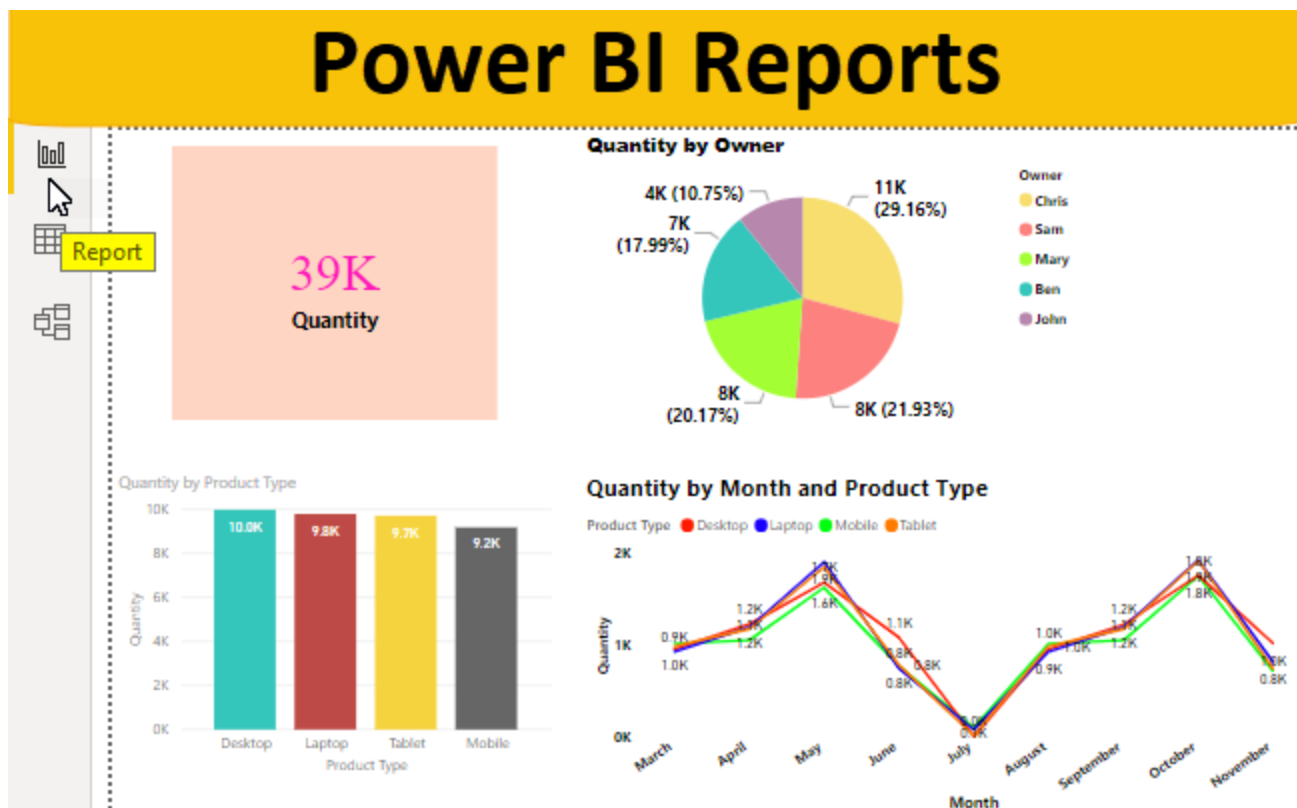
Capacity-based offering that includes flexibility to publish reports broadly across an enterprise without requiring recipients to be licensed individually per user. This provides greater scale and performance than shared capacity in the Power BI Service.

Power BI Visuals Marketplace

A marketplace of custom visuals and R-powered visuals.

Power BI Dataflow

A Power Query implementation in the cloud that can be used for data transformations to make a common Power BI Dataset, which can then be made available for report developers through Microsoft's Common Data Service. For example, it can be used as an alternative to doing transformations in SSAS, and may ensure that several report developers use data that has been transformed in a similar way.



Components of Power Bi:

1.Power Query

Power Query is the data transformation and mash up the engine. It enables you to discover, connect, combine, and refine data sources to meet your analysis need. It can be downloaded as an add-in for Excel or can be used as part of the Power BI Desktop.

2.Power Pivot :

Power Pivot is a data modeling technique that lets you create data models, establish relationships, and create calculations. It uses Data Analysis Expression (DAX) language to model simple and complex data.

3.Power View:

Power View is a technology that is available in Excel, Sharepoint, SQL Server, and Power BI.

It lets you create interactive charts, graphs, maps, and other visuals that bring your data to life.

It can connect to data sources and filter data for each data visualization element or the entire report.

4.Power Map:

Microsoft's Power Map for Excel and Power BI is a 3-D data visualization tool that lets you map your data and plot more than a million rows of data visually on Bing maps in 3-D format from an Excel table or Data Model in Excel. Power Map works with Bing maps to get the best visualization based on latitude, longitude, or country, state, city, and street address information.

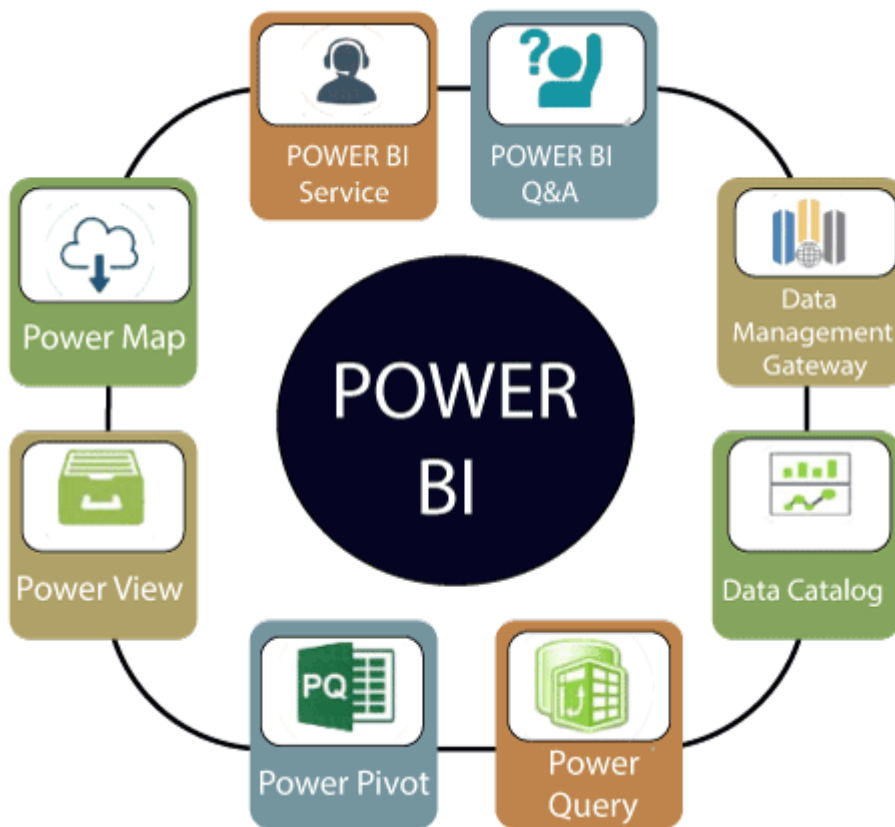
5.Power Bi Desktop:

Power BI Desktop is a development tool for Power Query, Power Pivot, and Power View.

With Power BI Desktop, you have everything under the same solution, and it is easier to develop BI and data analysis experience.

6.Power Q&A:

The Q&A feature in Power BI lets you explore your data in your own words. It is the fastest way to get an answer from your data using natural language. An example could be what was the total sales last year? Once you've built your data model and deployed that into the Power BI website, then you can ask questions and get answers quickly.



Practical Uses of Power BI

Microsoft Power BI will help you solve this problem with the help of a powerful business intelligence tool that mainly stresses on Visualization. Microsoft Power BI is a fundamental programming framework for organizations with huge amounts of disparate data developed during normal business operations. Power BI has numerous **uses in building a software** system. Let's see the practical uses of Power BI below:

1. Visualization = Inbuilt feature
2. Server-Level Data Management
3. Analytics With Internal Software Systems
4. Provide complex data within software and apps
5. Streamline Organizational Processes
6. Visualize Details Easily
7. Enhance the marketing.
8. Real-Time look at the company's financial performance
9. Create Consistent Reporting Standards

Final project

Problem Statement:

Create a classification model to predict whether CREDIT RISK is good or bad.

CONTEXT:

Financial institution, is interest is to know the potential financial whereabouts of the customers in order to determine whether the credit risk associated with them is good or bad. The data set could be used to predict if the customer could be given credit. Many features require data cleaning.

After that, we will use two data sets that emulate real credit applications on business values.

Dataset:

For modelling probability of default we generally have two primary types of data available:

Application data:

Which is data that is directly tied to the loan application like loan grade.

Behavioral data:

Which describes the recipient of the loan, such as employment length.

The data will use for our predictions of probability of default includes a mix.

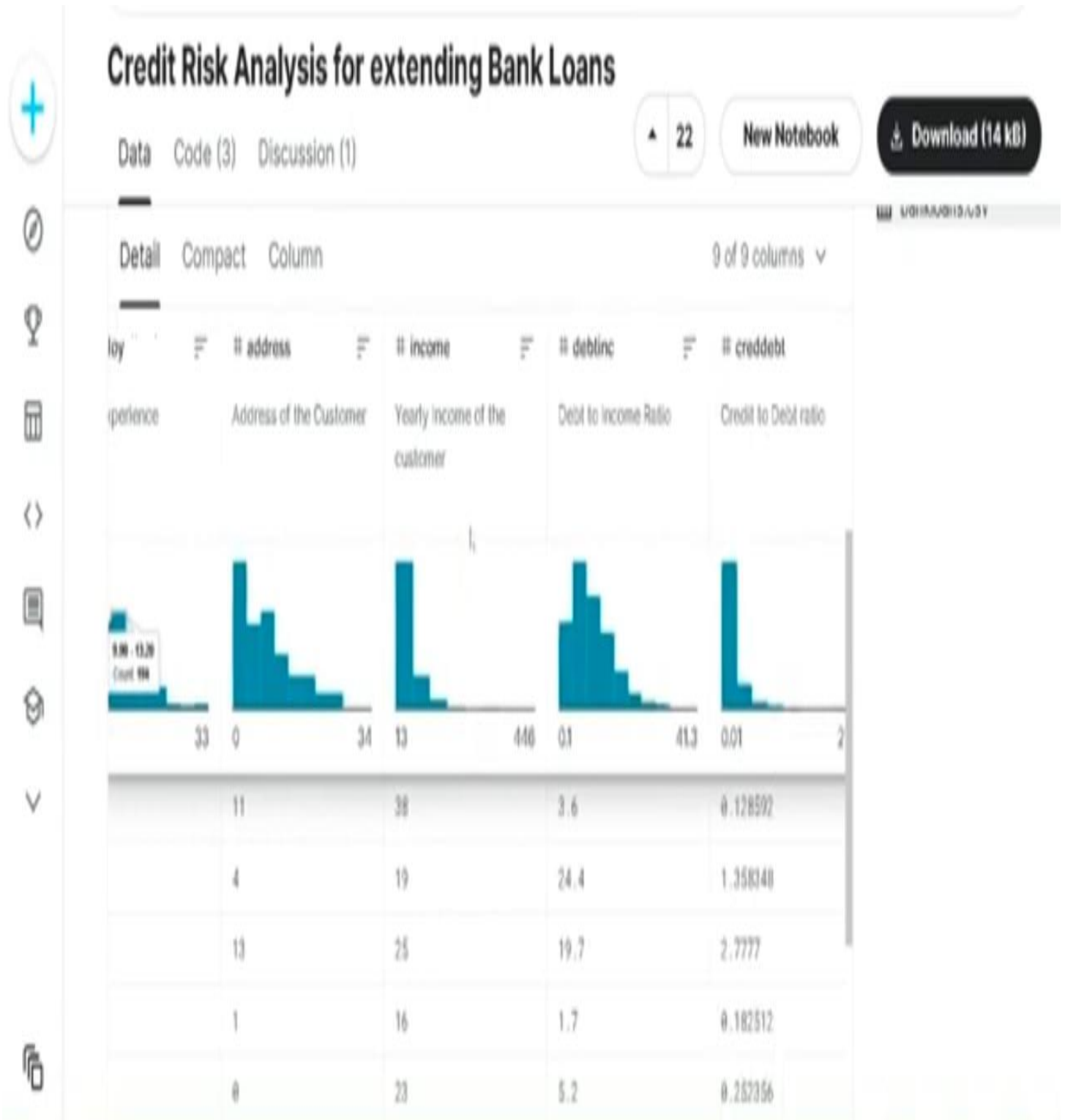
This is important because application data alone is not as good as application and behavioral data together.

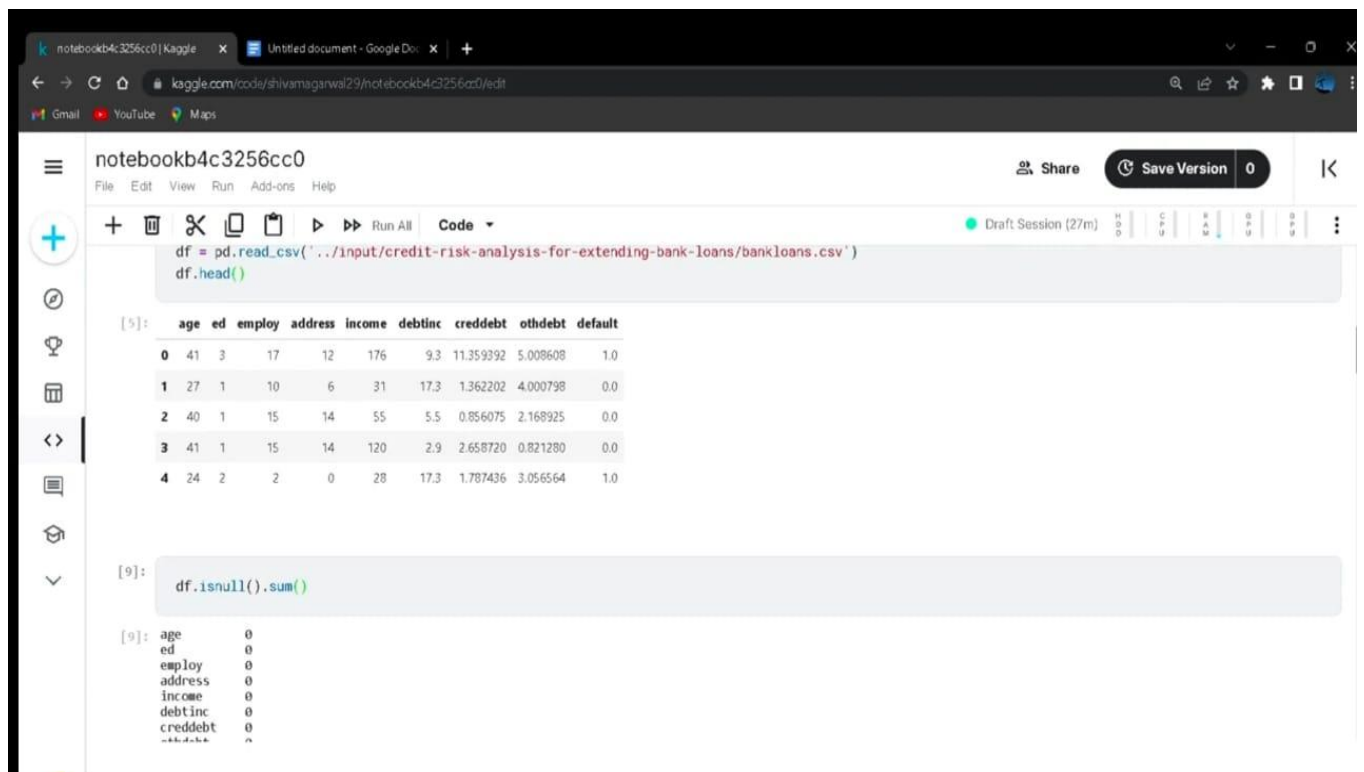
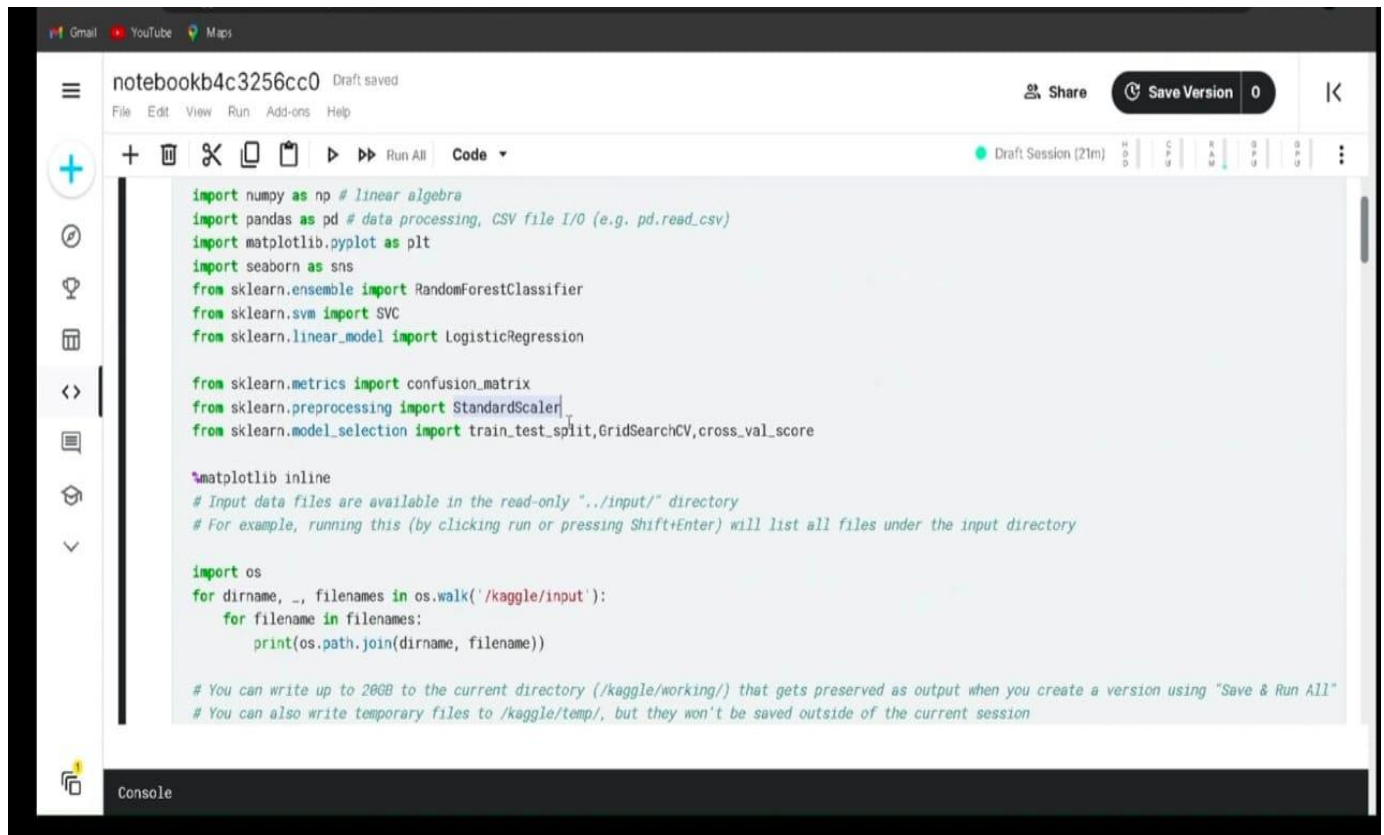
Credit Risk Analysis for extending bank loans:

Credit risk is perhaps one of the most 'classic' applications for predictive modelling, to predict whether or not credit extended to an applicant will likely

result in profit or losses for lending institution.

Solution:





Gmail
YouTube
Maps

notebookb4c3256cc0
Draft saved
Share
Save Version 0

File Edit View Run Add-ons Help

+
Run All
Code
Draft Session (13m)

```

out[0]:
default      0
dtype: int64

[7]:
df.value_counts()

[7]:
age  ed  employ  address  income  debtinc  creddebt  othdebt  default
20  1  4      0      14    9.7    0.200984  1.157016    1.0      1
39  1  10     4      31    4.8    0.184512  1.303488    0.0      1
    0      8      39    7.9    1.066026  2.014974    0.0      1
    2     15     22   23.1    1.915914  3.166086    1.0      1
    4      9     38    6.5    1.178190  1.291810    0.0      1
...
30  2   8      4     56    6.4    0.333312  3.250688    0.0      1
    10     4     22   16.1    1.409716  2.132284    0.0      1
    12      9     68   20.1    2.856612  10.811388    0.0      1
    56     1    11    20    59    15.0    4.672800  4.177200    0.0      1
length: 700, dtype: int64

+ Code + Markdown

[8]:
df = df.dropna()

```

Console

notebookb4c3256c0 | Kaggle

Untitled document - Google Docs

kaggle.com/code/shivamagarwal29/notebookb4c3256c0/edit

notebookb4c3256c...

File Edit View Run Add-ons Help

Share Save Version 0

56 1 11 20 98 7.2 2.935296 4.120104 0.0 1
59 15.0 4.672800 4.177200 0.0 1
Length: 700, dtype: int64

[8]:

```
df = df.dropna()
```

fig,ax=plt.subplots(figsize=(20,10))
sns.lineplot(x='age',y='income',data=df,ax=ax)

+ Code + Markdown

Data

+ Add Data

Input

credit-risk-analysis-for-extending
bankloans.csv

Output (60KB / 19.5GB)

/kaggle/working

Settings

ACCELERATOR

GPU T4 x2

Quota: 00:00 / 33 hrs

LANGUAGE

Console

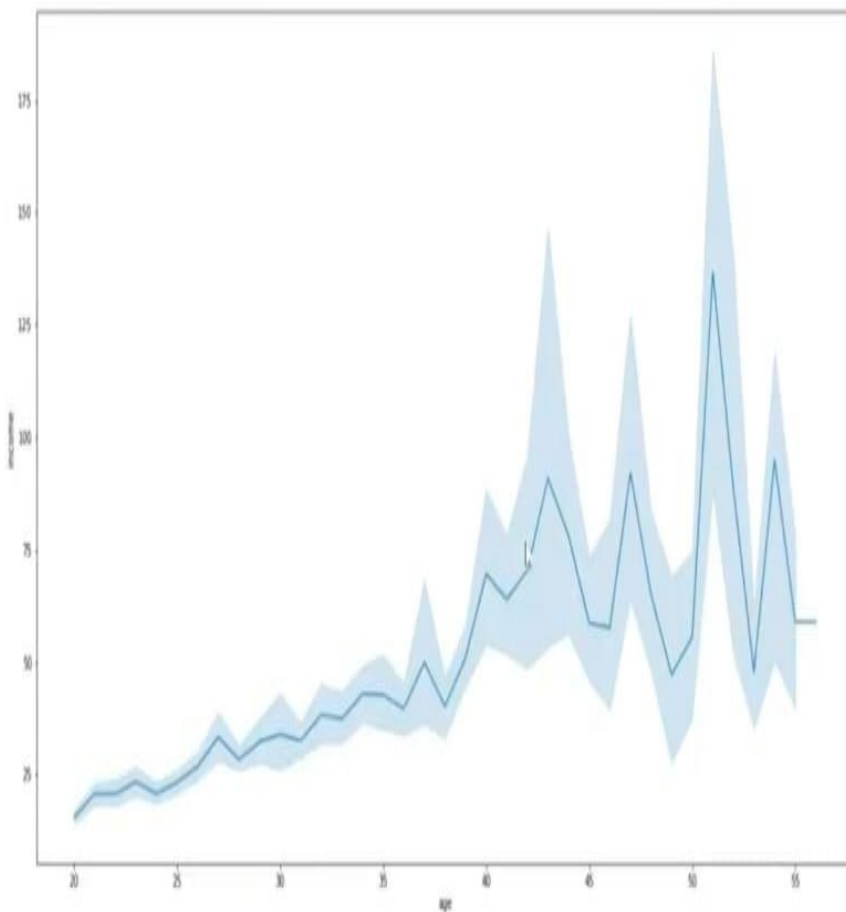
notebookb4c3256c...

Share

Save Version 0

>|

File Edit View Run Add-ons Help



+ Code

+ Markdown

Data

+ Add Data



Input

credit-risk-analysis-for-extending

bankloans.csv

Output (60KB / 19.5GB)

/kaggle/working

Settings

ACCELERATOR

GPU T4 x2

Quota: 00:00 / 33 hrs

LANGUAGE



Console

notebookb4c3256cc0 Draft saved

File Edit View Run Add-ons Help

Share Save Version 0

Draft Session (18m)

age

```
[12]: df['default'].value_counts()
```

```
[12]: 0.0    517  
      1.0    183  
      Name: default, dtype: int64
```

```
[13]: x=df.drop(['default'],axis=1)  
      y=df['default']
```

```
xtrain,xtest,ytrain,ytest
```

+ Code + Markdown

Console

notebookb4c3256cc0

File Edit View Run Add-ons Help

Share Save Version 0

Draft Session (27m)

Random Forest

```
rfc = RandomForestClassifier(n_estimators=200)
```

```
[19]: rfc.fit(xtrain,ytrain)
```

```
[19]: RandomForestClassifier(n_estimators=200)
```

```
[20]: rfc.score(xtest,ytest)
```

```
[20]: 0.8
```

Console

Gmail YouTube Maps

notebookb4c3256cc0

File Edit View Run Add-ons Help

Share Save Version 0

+ Run All Code ▾

Draft Session (35m) H D C P R G O P O

[25]:

model = GridSearchCV(sv,{
 'C':[0.1,0.2,0.4,0.8,1.2,1.8,4.0,7.0],
 'gamma':[0.1,0.4,0.8,1.0,2.0,3.0],
 'kernel':['rbf','linear']
},scoring='accuracy',cv=10)

[26]:

model.fit(xtrain,ytrain)

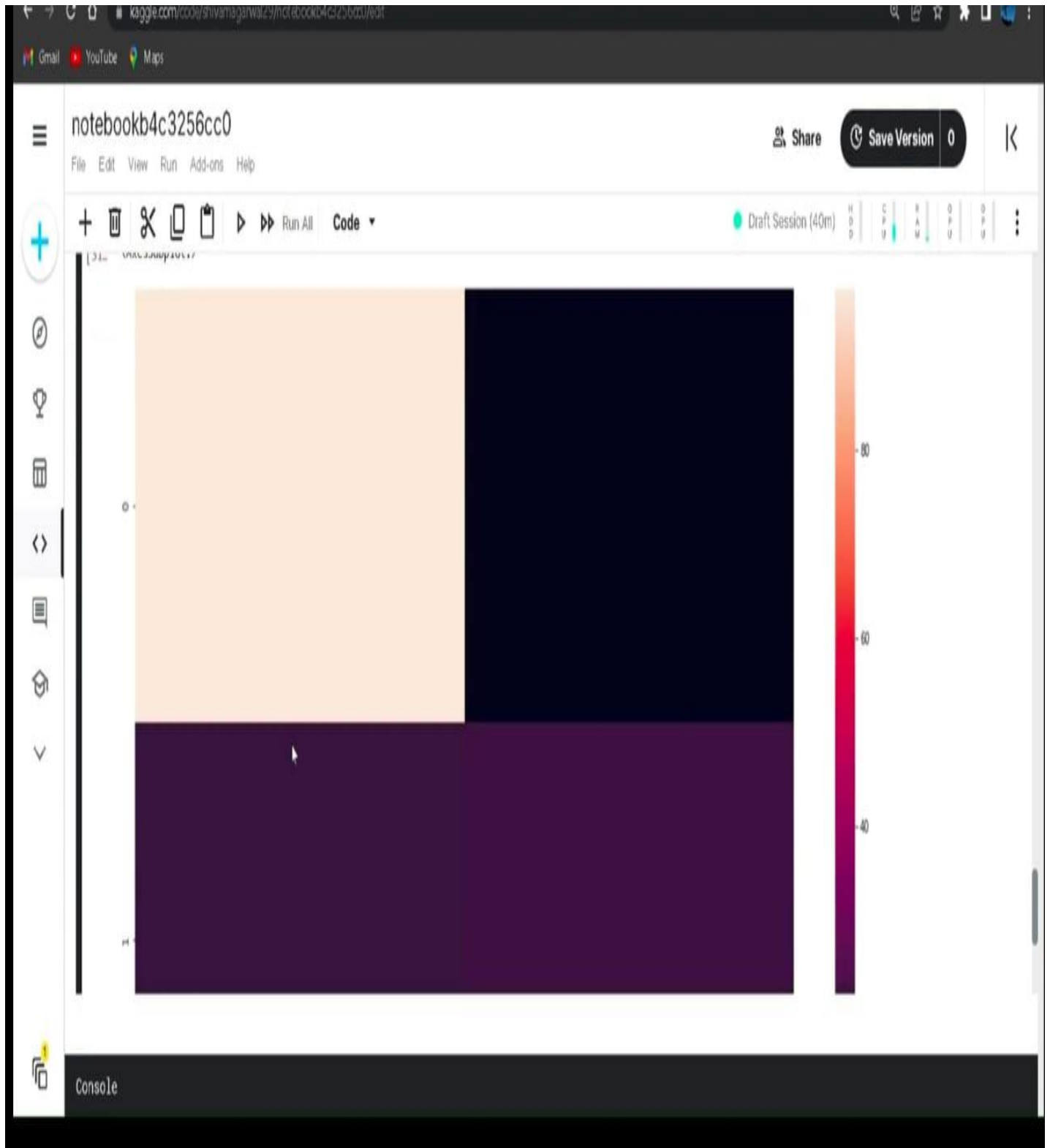
[26.. GridSearchCV(cv=10, estimator=SVC(),
 param_grid={'C': [0.1, 0.2, 0.4, 0.8, 1.2, 1.8, 4.0, 7.0],
 'gamma': [0.1, 0.4, 0.8, 1.0, 2.0, 3.0],
 'kernel': ['rbf', 'linear']},
 scoring='accuracy')

I

+ Code + Markdown

Console

OUTPUT:



RESULT:

By analyzing the above graph we can predict that the credit risk easily. when the colors are

Dark that means the occurrence of risk is high.

When there is a light color the occurrence of risk is low

Conclusion

- To sum up, DS is an interdisciplinary science, but without a clear definition. It can be defined as a set of skills from computer science, statistics,
- It definitely requires some Research qualities, but also Domain Expertise.
- Different Data Scientists used to focus or specialize in one area of experience.
- Data and the capability to extract useful knowledge from data, shouldbe regarded as key strategic assets.
- Extracting useful knowledge from data to solve business problems must be application like loan ground.

