

What is Artificial Intelligence

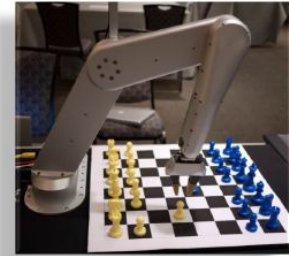
- Artificial Intelligence refers to intelligence displayed by machines that simulates human and animal intelligence.



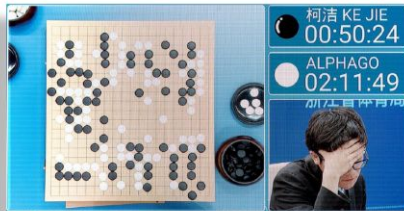
Self-driving cars



Siri(iPhone)



Chess



Google's AlphaGo

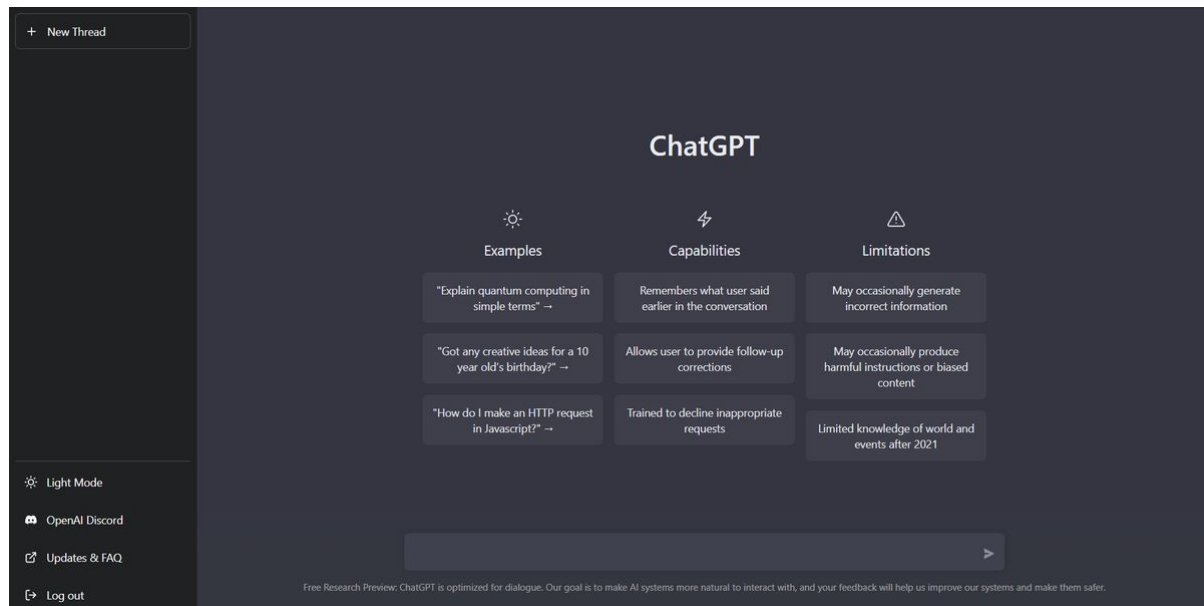


Amazon ECHO



*Concierge robot from IBM
Watson*

What is Artificial Intelligence



Data Facilitates in Recommendations

Amazon collects data from users and recommends the best products according to the user's buying/shopping pattern.

amazon

Added to Cart

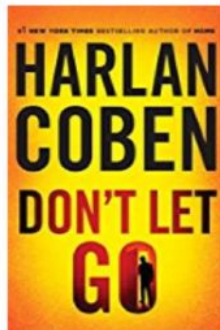
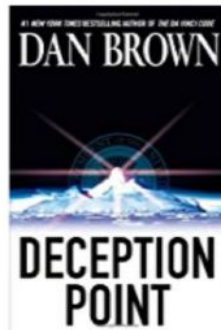
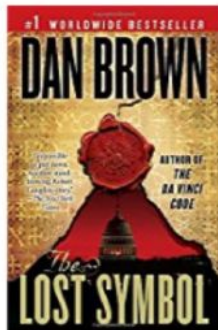
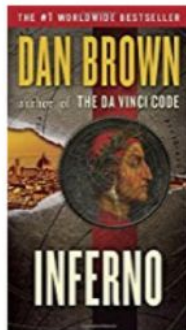
Cart subtotal (1 item): **\$17.96**

To qualify for **FREE Shipping**, add **\$7.04** of eligible items. [Details](#)

Cart

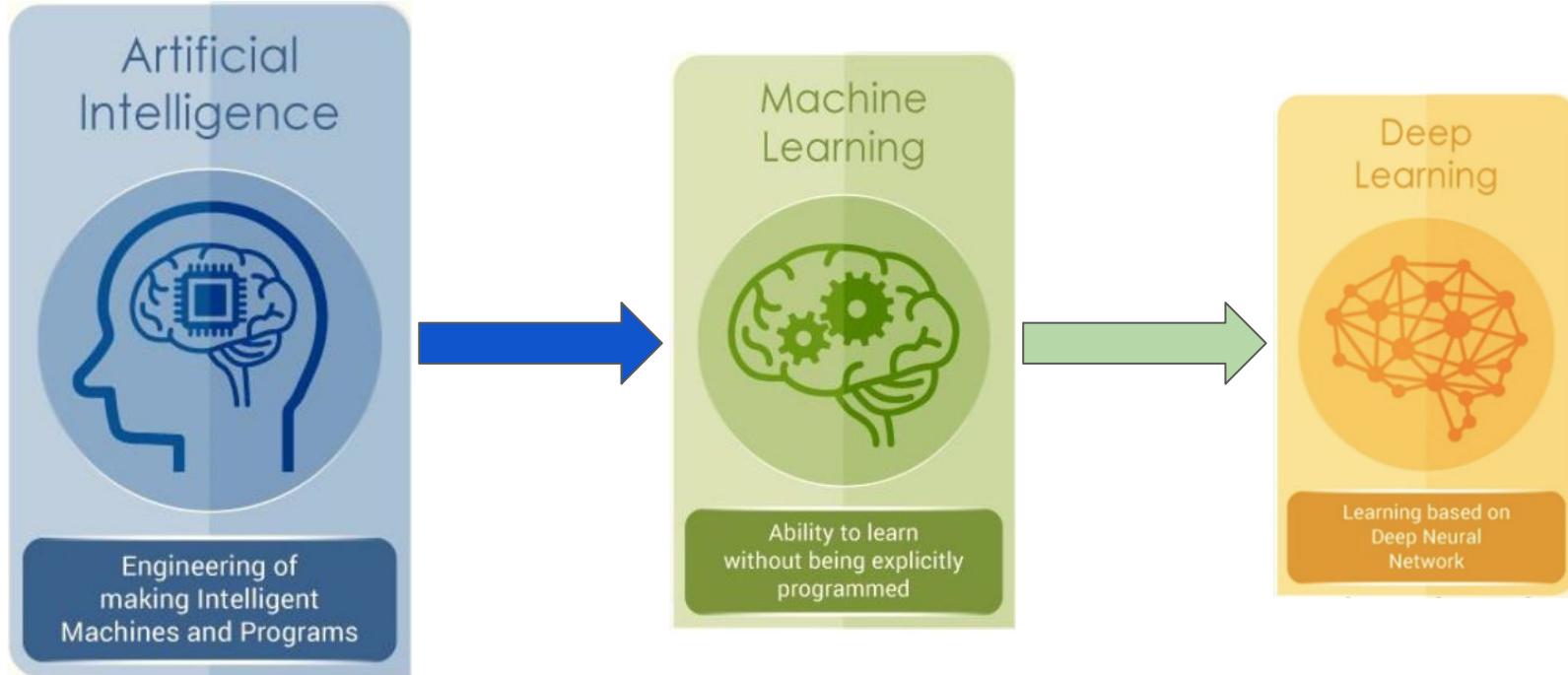
Proceed to checkout (1 item)

Customers who bought *Origin: A Novel* also bought



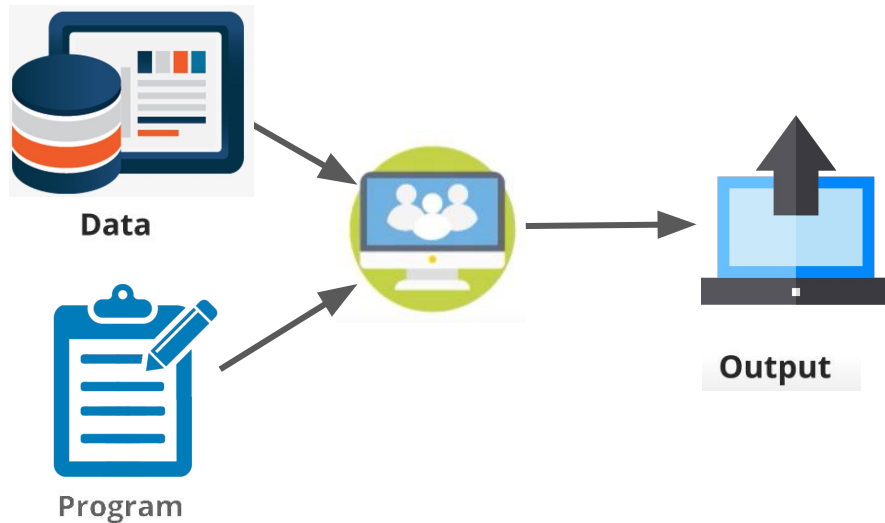
Artificial Intelligence

- The capability of Artificial Intelligence systems to learn by extracting patterns from data is known as Machine Learning.



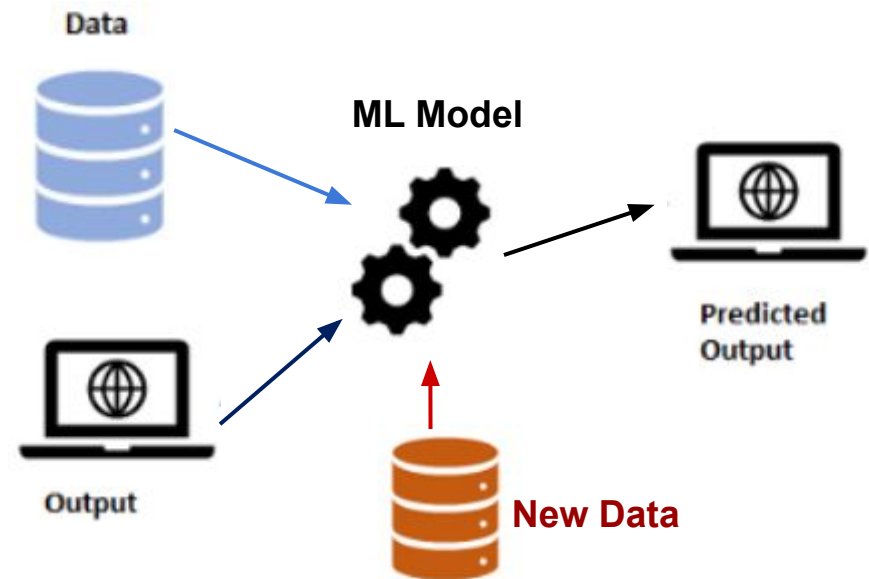
Traditional Programming

Traditional Programming: You code the behavior of the program

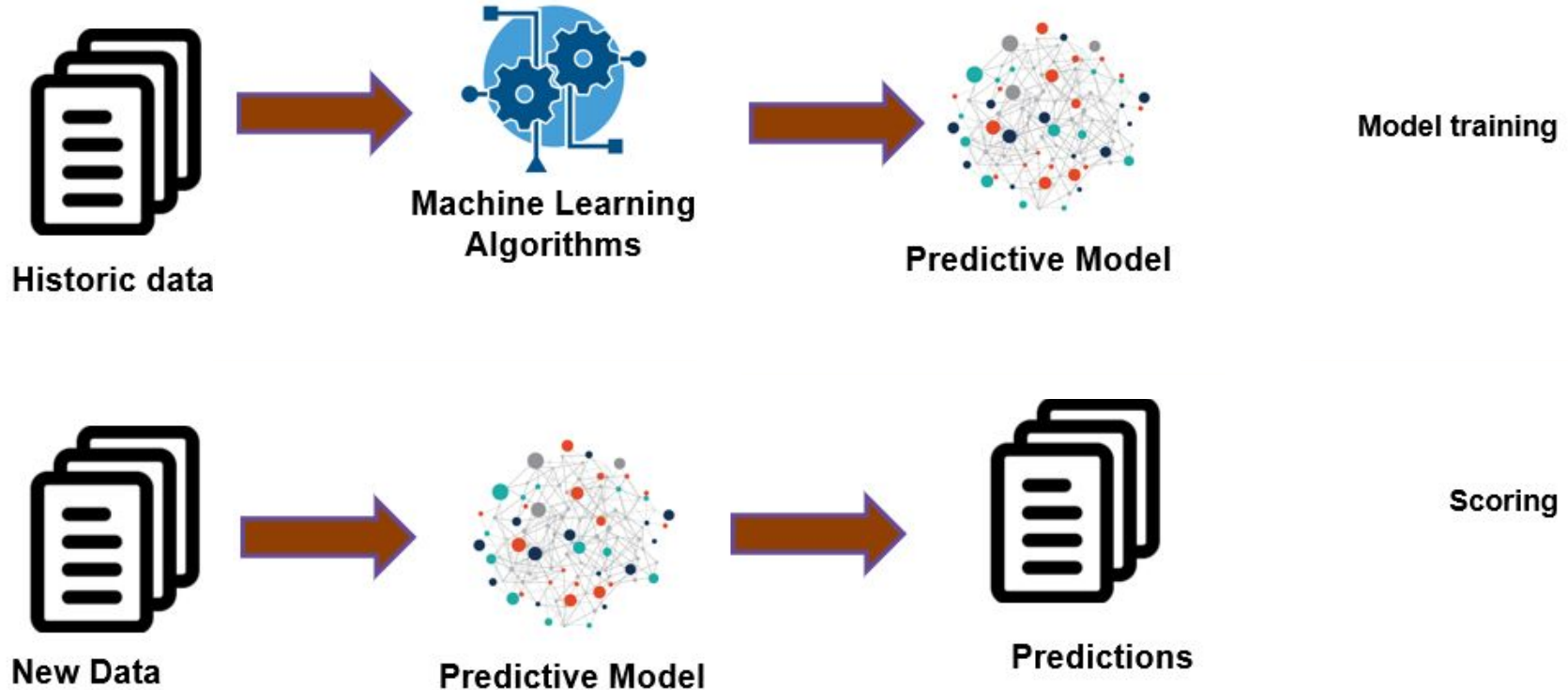


Machine Learning

Machine Learning: You leave a lot of that to the machine to learn from data

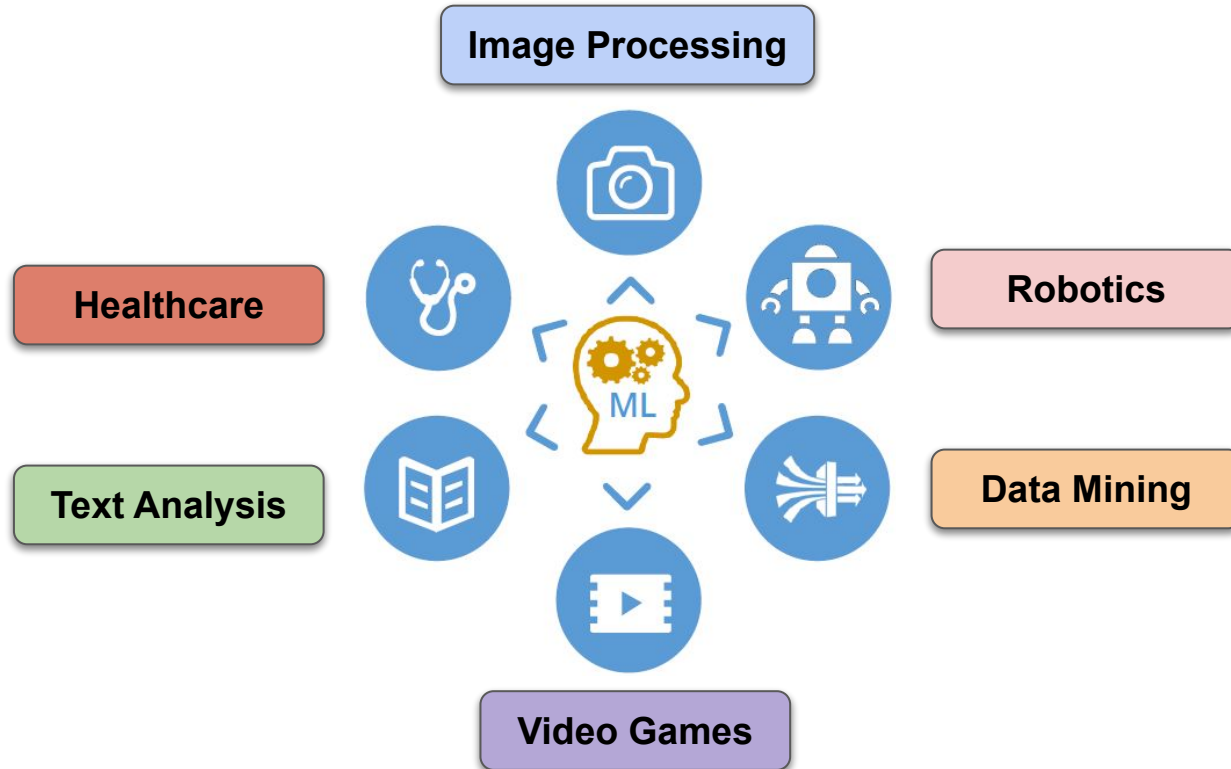


Machine Learning Predictive Model

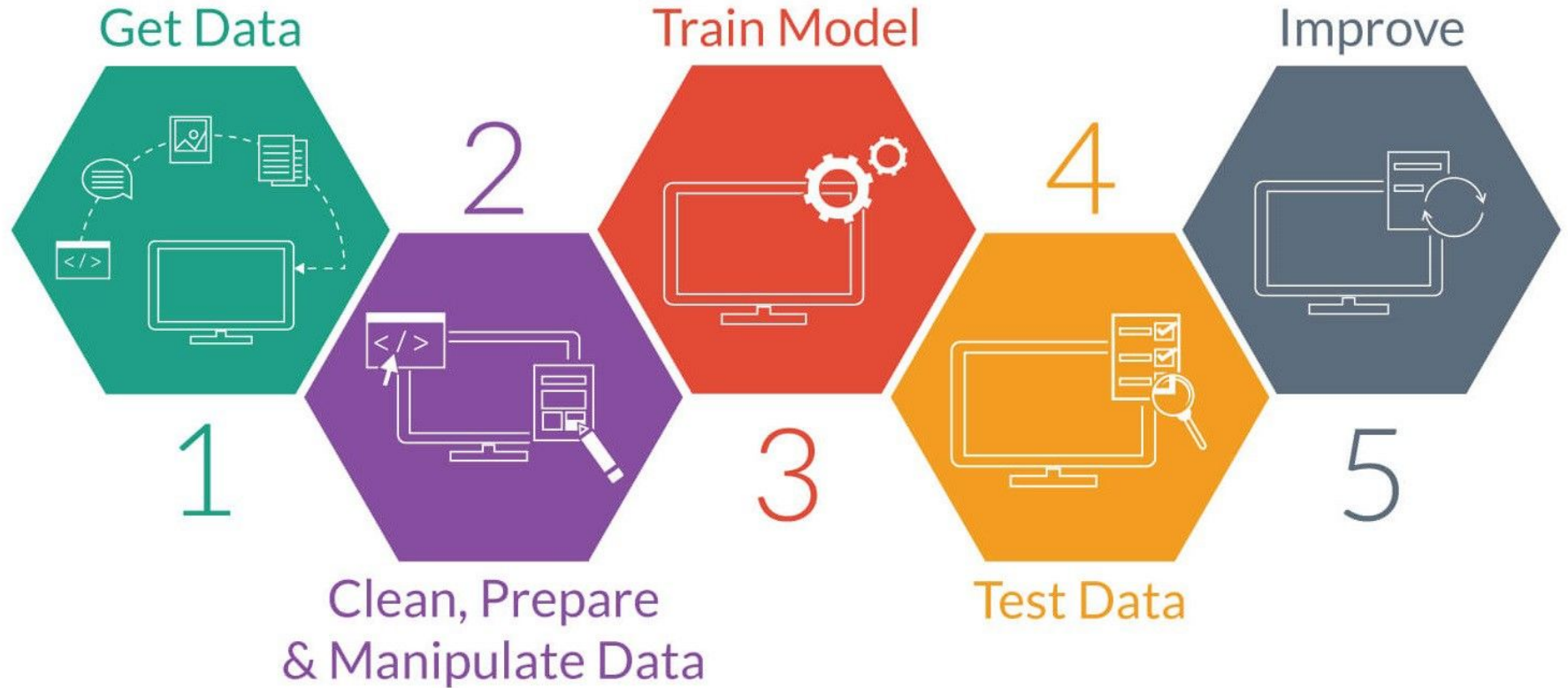


Applications of Machine Learning

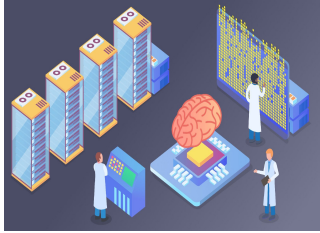
- The capability of Artificial Intelligence systems to learn by extracting patterns from data is known as Machine Learning.



Machine Learning Cycle:



Steps in Machine Learning



1. Data Gathering



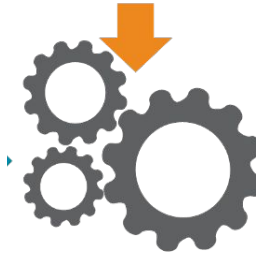
2. Data Cleaning



3. Data Visualization



Predictive Model



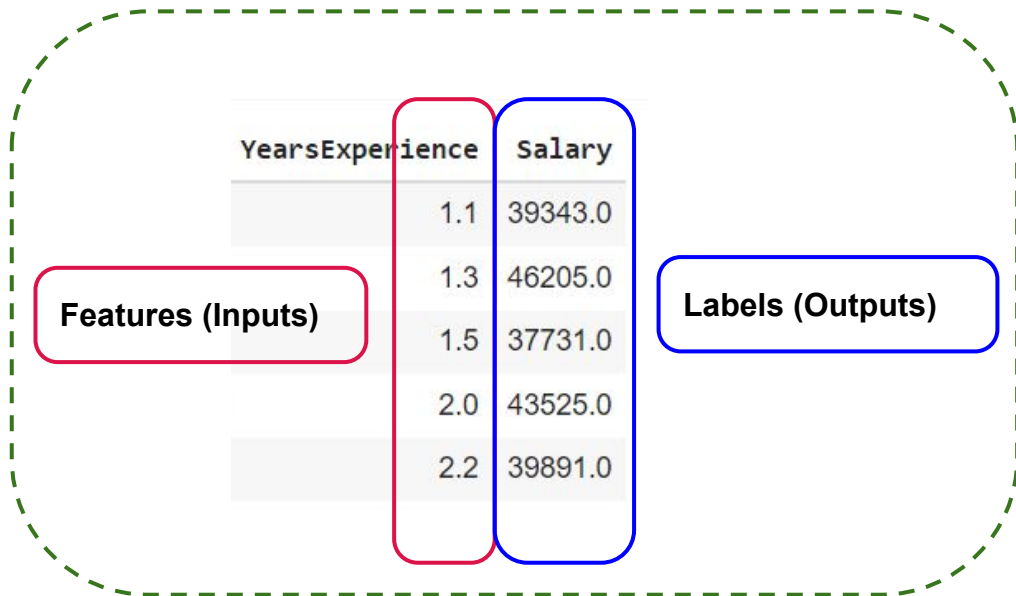
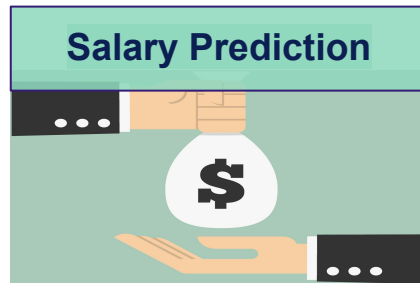
5. Training & Testing



4. Data Processing



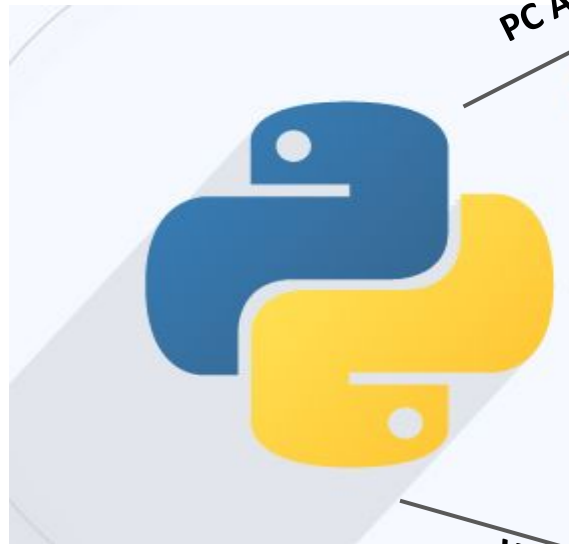
Predictive Model



ML Model



Predictive Model



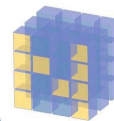
PC Applications

Top 5 Python GUI Libraries



Machine Learning

matplotlib



NumPy



Pandas



Seaborn

Web Development

CherryPy

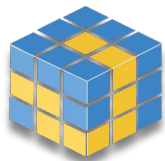


python™

django

Pyramid™

Python Packages

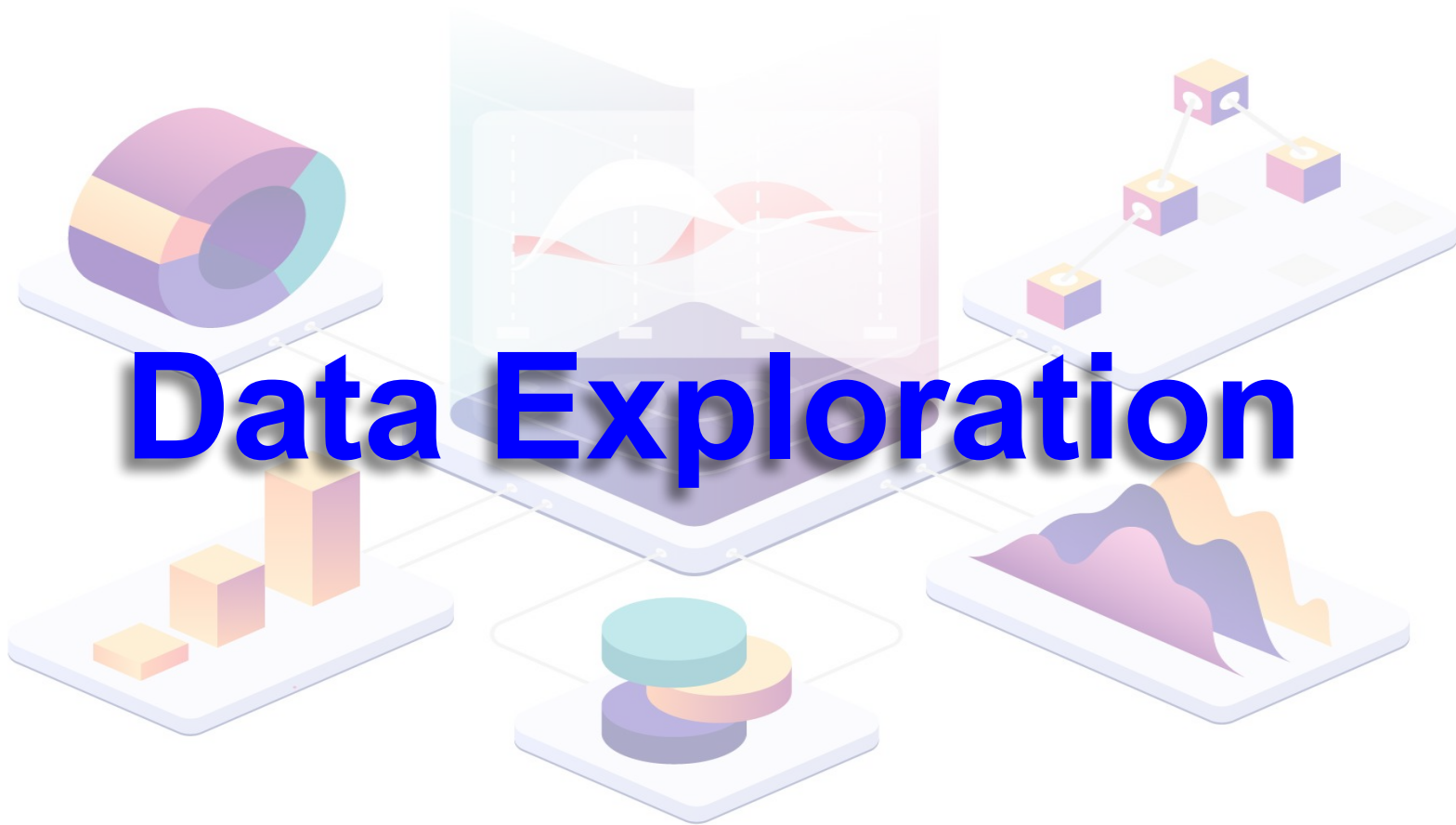


NumPy

matplotlib



Data Exploration



Types of Data



Ex:

- Salary
- Age
- Weight

Ex:

- Male / Female
- Yes / No
- Rating



Numerical Data

Weight

SAT Score

Continuous
Data

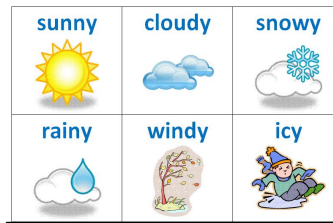
Discrete
Data

Data Changes with
No Limit

Data within some
numerical Range



Categorical:



Categorical Data

**Nominal
Data**

Ordinal Data

**Categories without
mathematical
weightage**

**Data with
mathematical
weightage**



Pen



Pencil



Eraser



Cow



Dog



Cat



4



An open-source Python
Library

Used for
high-performance
data manipulation and
analysis

Data Types in Pandas

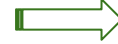


Series

- 1-Dimensional Homogeneous Data

Index	Data	
	0	254
	1	253
	2	0
	3	3
	4	2

A callout bubble labeled "Series" points to the data column of the table above.



DataFrame

- 2-Dimensional Heterogeneous Data

	Name	Symbol	Shares
0	Microsoft Corporation	MSFT	100
1	Google, LLC	GOOG	50
2	Tesla, Inc.	TSLA	150
3	Apple Inc.	AAPL	200
4	Netflix, Inc.	NFLX	80

Loading .csv File in Python Program



titanic Dataset

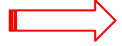
`pd.read_csv('titanic.csv')`



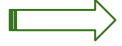
Python Code

```
import pandas as pd
Data = pd.read_csv("titanic.csv")
Data.head()
```

Data Accessing Methods in Pandas



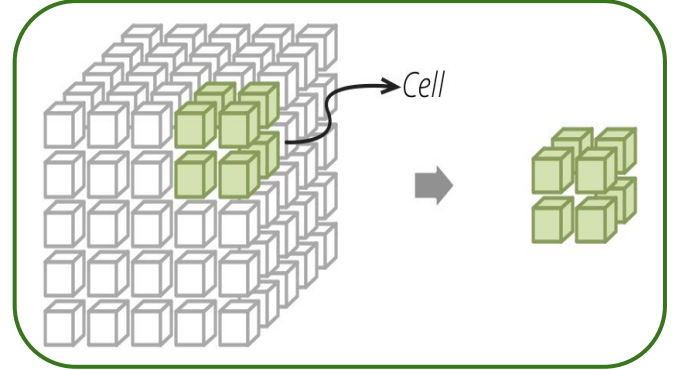
1. Indexing



2. Slicing



3. Filtering



Indexing Method



Accessing single column

`df[["<column_Name>"]]`

Python Code:

```
import pandas as pd
Data = pd.read_csv("titanic.csv")
print(Data["Name"])
```

Accessing Multiple columns

`df[["<column1>","<column2>,.."]]`

Python Code:

```
import pandas as pd
Data = pd.read_csv("titanic.csv")
Data[["PassengerId", "Name"]]
```

Indexing Method



.loc function



Access Rows

`df.loc["row1"]`

Ex:

```
import pandas as pd
dat = pd.read_csv("titanic.csv")
dat.head()
dat.loc[1]
```

Access Rows & Columns

`df.loc[["row1","row2"],["col1","col2"]]`

Ex:

```
import pandas as pd
dat = pd.read_csv("titanic.csv")
dat.head()
dat.loc[[1,2,3,4,5],["Name","Age"]]
```

Slicing Method



.iloc function

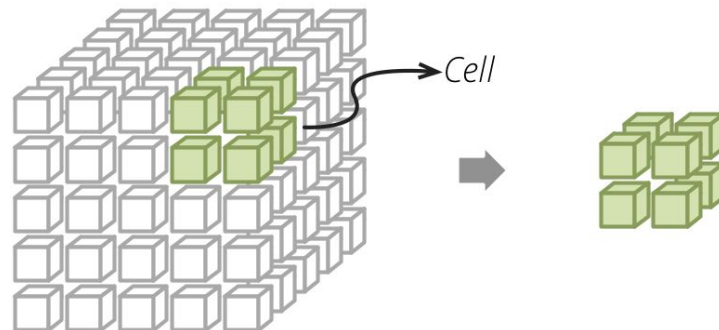


Access Rows & Columns using index range.

df.iloc[<row_range>, <col_range>]

Python Code

```
import pandas as pd
Data = pd.read_csv("titanic.csv")
Data.iloc[0:2,0:3]
```



	PassengerId	Survived	Pclass
0	892	0	3
1	893	1	3

Filtering

→ Filter the required data based on Logic.



Python Code

```
import pandas as pd
Data = pd.read_csv("titanic.csv")
Data[Data["Survived"]>0]
```

	PassengerId	Survived	Pclass	Name	Sex	Age
1	893	1	3	Wilkes, Mrs. James (Ellen Needs)	female	47.0
4	896	1	3	Hirvonen, Mrs. Alexander (Helga E Lindqvist)	female	22.0
6	898	1	3	Connolly, Miss. Kate	female	30.0
8	900	1	3	Abraham, Mrs. Joseph (Sophie Halaut Easu)	female	18.0
12	904	1	1	Snyder, Mrs. John Pillsbury (Nelle Stevenson)	female	23.0
...
409	1301	1	3	Peacock, Miss. Treasteall	female	3.0
410	1302	1	3	Naughton, Miss. Hannah	female	NaN
411	1303	1	1	Minahan, Mrs. William Edward (Lillian E Thorpe)	female	37.0
412	1304	1	3	Henriksson, Miss. Jenny Lovisa	female	28.0
414	1306	1	1	Oliva y Ocana, Dona. Fermina	female	39.0

Data Exploration Techniques

1. Identifying Unique Elements

→ `dat["PassengerId"].unique()`

2. Value Count

→ `dat["Sex"].value_counts()`

3. Null Value Check

→ `dat.isnull().sum()`

4. Drop Feature

→ `dat.drop("Sex", axis = 1)`

5. Feature Mean

→ `dat["Age"].mean()`

6. Feature Median

→ `dat["Age"].median()`

7. Feature Mode

→ `dat["Sex"].mode()`

Map Function in Pandas:

Syntax:

```
df[“<column_name”] = df[“<column_name”].map(<function_name>)
```

Python Code:

```
import numpy as np
import pandas as pd
data=pd.read_csv('employee.csv')
data.head()

def function(x):
    return x/10

data['DailyRate'] = data['DailyRate'].map(function)
data['DailyRate']
```

Apply Function in Pandas

Syntax:

```
df[["<col1", "<col2"]]= df[["<col1", "<col2"]].apply(<function_name>)
```

Python Code:

```
import numpy as np
import pandas as pd
data=pd.read_csv('employee.csv')
print(data.head())
```

```
def function(x):
    return x/10
```

```
data[['DailyRate', 'MonthlyRate']] = data[['DailyRate', 'MonthlyRate']].apply(function)
data[['DailyRate', 'MonthlyRate']]
```

Data Cleaning

Data Cleaning Methods



1. Dropping



Deleting the Rows or columns



2. Imputing



Imputing the missing value by statistical Method (Mean, Median & Mode)

Dropping Method

→ Deleting the Rows or columns

	column_a	column_b	column_c	column_d	column_e
0	1.0	1.2	a	True	1
1	2.0	1.4	NaN	True	2
3	4.0	6.2	d	None	4
5	NaN	1.1	NaN	True	5
6	6.0	4.3	d	False	NaN

**In Row,
Missing values > 30%
Delete Row**

**In Column,
Missing values > 30%
Delete Column**



When should we delete?