

# **Highradius (Highway to Highradius) Internship Program**

**Highradius**

A Project Report

Submitted in partial fulfillment of the requirement for the award of degree of

**B. Tech in Computer Science and Engineering (CSE)**

Submitted to

**LOVELY PROFESSIONAL UNIVERSITY**

**PHAGWARA, PUNJAB**



**L** OVELY  
**P** ROFESSIONAL  
**U** NIVERSITY

**From 28/01/2022 to 14/04/2022**

**Submitted By**

**Name: Radharaman sharma**

**Registration no.: 11910481**

## **STUDENT DECLARATION**

**To whom so ever it may concern**

I, **Radharaman sharma,11910481** hereby declare that the work done by me on “**Highradius (Highway to Highradius) Internship Training Program**” from **Jan, 2022** to **Apr, 2022**, is a record of original work for the partial fulfilment of the requirements for the award of the degree, **B. Tech in Computer Science and Engineering (CSE)**.

**TRAINING CERTIFICATE**

## Product and Engineering

Product Essentials Program

This is to certify that **Radharaman Sharma** has successfully completed the Highway to HighRadius Internship Program from **28th January 2022** to **13th April 2022**, where he/she built and deployed an AI Enabled Fintech B2B Cloud Application.

During this project, he/she was involved in creating a full stack web-based product thereby developing a deep understanding of all aspects of product development such as identifying appropriate user requirements, designing a great user experience and building appropriate data models and machine learning models along with relevant UI components and backend design.

*Neha Srivastava*

Neha Srivastava  
AVP, People & Culture  
HighRadius





# ACKNOWLEDGEMENT

I would take this opportunity to express my sincere thanks and gratitude to my university as well as our Training Mentors Ms.shruti ,Mr.vinay for their vital support and guidance throughout the training.

I would also like to extend my gratitude and thanks to our assistant instructor Mr.Vishnu who has helped me in successful completion of my project.

I am thankful and fortunate enough to get constant encouragement, support and guidance from all teaching staff of **“SCHOOL OF COMPUTER SCIENCE AND ENGINEERING”** which helped me in successfully completing this project.

Radharaman sharma

11910481

Lovely Professional University

Phagwara, Punjab

## CONTENTS

<b>S. No.</b>	<b>Title</b>	<b>Page</b>
1	Declaration	2
2	Certificate	3
3	Acknowledgement	4
4	Contents	5
5	About the Organization	6
6	About the Project Undertaken	8
7	Chapter 1: Introduction	9
8	Chapter 2: Solution & Implementation	12
9	2.1 Module1: Machine Learning	13
10	2.2 EDA	28
11	2.3 Module2: Web application Development	37
12	2.4 JDBC	43
13	2.5 Front-end Development	48
12	Chapter 3: Conclusion & Discussions	62
13	3.1 Learning Outcomes	63
14	3.2 Conclusion	64
15	References	65

## About the Organization

HighRadius, a firm founded by an Indian-origin entrepreneur, is the latest unicorn in the United States. ICONIQ Capital has invested \$125 million in Houston-based HighRadius in a series B round of funding. Sashi Narahari, an Indian entrepreneur, launched the company, which has offices in Hyderabad, London, and Amsterdam.

It is a Fintech enterprise Software-as-a-Service (SaaS) company that leverages Artificial Intelligence-based Autonomous Systems to help companies automate Accounts Receivable and Treasury processes. The HighRadius® Integrated Receivables platform reduces cycle times in your order-to-cash process by automating receivables and payments processes across credit, electronic billing and payment processing, cash application, deductions, and collections

HighRadius Credit Software automates the credit management process, enabling credit managers to make highly-accurate credit decisions 2X faster and enable faster customer onboarding with 4 primary components: configurable online credit application, customizable credit scoring engines, credit agency data aggregation engine, and collaborative credit management workflow. Along with that, there are a lot of key features that should definitely be explored some of which are online credit application, credit information aggregation, automated credit scoring & risk assessment, credit management workflows, approval workflows, and automated bank & trade reference checks. The result is faster customer onboarding, better internal collaboration, higher customer satisfaction, more targeted periodic reviews, and lower credit risk across the company's customer portfolio.

**MISSION:**

We empower our customers to be able to work more accurately and efficiently. Efficiency and productivity enhancements are central to the value HighRadius provides to our customers.

**VISION:**

Vision of the organization is to power career success for every member of the global workforce as their trusted lifelong learning partner.

**GROWTH:**

Despite a challenging 2020, HighRadius registered a 47% growth in Contracted ARR (Annual Recurring Revenue) compared to the previous year.

As of today, HighRadius has grown to 600 customers, including over 200 customers from Forbes Global 2000.

The HighRadius AI-powered Integrated Receivables Platform processed \$2.23 trillion in annual receivables in 2020. This is the equivalent of France's GDP, and transaction volume increased by 74% over the previous year.

In March 2021, the company raised \$300 million in a Series C financing, bringing its valuation to \$3.1 billion in a year.

In comparison to the previous year, Contracted ARR (Annual Recurring Revenue) increased by 47%. Since the onset of the epidemic, HighRadius has seen a triple-digit rise of 143 percent in Europe (January 2020).



## **About the Project Undertaken**

### **Objective:**

**The project's objective was to build an AI-Enabled FinTech B2B Invoice Management Application.**

**The project was divided into two parts:**

- Machine Learning (For Due Date Prediction)
- Full Stack Web Application Development (ReactJs, Java, Servlets, JDBC)

### **Objective of ML part:**

As part of our internship project, our objective was to create a web application to assist those working in Accounts Receivable departments with their daily tasks. We were required to build a web application that allows Account Receivable users to: Get account-level analytics to quickly display and comprehend data- EDA and Feature Engineering. View the invoice from a certain buyer's various fields/attributes. And also get an estimate for when the invoice will be paid.

Pre-process the invoice data via data pre-processing. View data from various buyers' invoices.

### **Objective of Web Application Development:**

The Web Application Development internship project's goal was to create a full-stack invoice management application using Reacts, JDBC, Java, and Servlets. In predefined templates, user should be able to delete data from chosen rows, add and edit data in the grid's editable fields. On the bills, user should be able to perform searching operations. Receivables Dashboard should be friendly. Grids can be used to visualize data.



# CHAPTER-1

## INTRODUCTION

## **INTRODUCTION**

The B2B world is not the same as the B2C or C2C worlds. Businesses work on a credit basis with one another. When a buyer company orders products from a seller company, the seller company sends an invoice to the buyer company. This products invoice includes information such as the details of the goods purchased and when they should be paid. In accounting, this is referred to as "Accounts Receivable."

Money due by organizations to the company on the sale of goods or services on credit is represented in accounts receivable. In the majority of business entities, accounts receivable are normally handled by creating an invoice, mailing it or delivering it electronically to the client, who is then required to pay it within a predetermined window of time, known as the credit terms or payment terms.

At different times, the seller business interacts with multiple businesses and sells things to all of them. As a result, the seller company must keep track of the total amount it owes all of the buyers. This entails keeping track of all of the buyers' invoices. Each invoice will include fields such as the payment due date, invoice date, invoice amount, baseline date, and so on.

Before the due date, the buyer's company must clear its outstanding balance. However, invoices are not always cleared, that is, paid in full by the due date, in real-world settings. The payment date refers to the day on which a customer makes a payment on an invoice.

The problem statement was to build a prediction model which could predict the due date of bills for customer. And also to build a UI using react and Java , java servlet and JDBC for the backend.

The screenshot displays a web application interface for "ABC Products" by "highradius". The interface includes a sidebar with navigation options: PREDICT, ANALYTICS VIEW, ADVANCE SEARCH, and a refresh button. A search bar labeled "Search Customer Number" is present, along with buttons for ADD, EDIT, and DELETE. The main content area features a table with 15 columns: Sl No., Business Code, Customer Number, Clear Date, Business Year, Document Id, Posting Date, Document Create Date, Due in Date, Invoice Currency, Document Type, Posting Id, Total Open Amount, and Baseline Create Date. The table contains 8 rows of data. At the bottom, there is a footer with "Privacy Policy | 2022 Highradius Corporation. All right reserved" and a pagination control showing "Rows per page: 10" and "1-10 of 2000".

Sl No.	Business Code	Customer Number	Clear Date	Business Year	Document Id	Posting Date	Document Create Date	Due in Date	Invoice Currency	Document Type	Posting Id	Total Open Amount	Baseline Create Date
1	U001	200769623	2020-02-11	2020-01-01	1930438491	2020-01-26	2020-01-25	2020-02-10	USD	RV	1	54273.28	2020-01-26
2	U001	200980828	2019-08-08	2019-01-01	1929646410	2019-07-22	2019-07-22	2019-08-11	USD	RV	1	79656.6	2019-07-22
3	U001	200792734	2019-12-30	2019-01-01	1929873765	2019-09-14	2019-09-14	2019-09-29	USD	RV	1	2253.86	2019-09-14
4	CA02	140105686	0000-00-00	2020-01-01	2960623488	2020-03-30	2020-03-30	2020-04-10	USD	RV	1	3299.7	2020-03-31
5	U001	200769623	2019-11-25	2019-01-01	1930147974	2019-11-13	2019-11-13	2019-11-28	USD	RV	1	33133.29	2019-11-13
6	CA02	140106181	2019-12-04	2019-01-01	2960581231	2019-09-20	2019-09-20	2019-10-04	CAD	RV	1	22225.84	2019-09-24
7	U001	200769623	2019-11-12	2019-01-01	1930083373	2019-11-01	2019-10-31	2019-11-16	USD	RV	1	7358.49	2019-11-01
8	U001	200744019	0000-00-00	2020-01-01	1930659387	2020-03-19	2020-03-18	2020-04-03	USD	RV	1	11173.02	2020-03-19

User interface of Web Application project

CHAPTER-2  
**SOLUTION / IMPLEMENTATION**

## Module1: Machine Learning

### Machine Learning Module:

First Module of the training was Machine Learning which we implemented in our project. ML module included different sub-modules like: Basics of python language, libraries for data analysis: NumPy, Pandas and related functions and their functionalities.

### Python Module

Python is a very simple coding language that uses a very familiar language to code. It uses indentation to define blocks of code and they need to be consistent throughout the block.

#### Operators in Python:

Operator	Description	Example	Operator	Description	Example
+	Addition	2 + 4 == 6	,	Comma	range(0, 10)
-	Subtraction	2 - 4 == -2	:	Colon	def X():
*	Multiplication	2 * 4 == 8	.	Dot	self.x = 10
**	Power of	2 ** 4 == 16	=	Assign equal	x = 10
/	Division	2 / 4.0 == 0.5	;	semi-colon	Print("hi"); print("there")
//	Floor division	2 // 4.0 == 0.0	+=	Add and assign	x = 1; x += 2
%	String interpolate or modulus	2 % 4 == 2	-=	Subtract and assign	x = 1; x -= 2
<	Less than	4 < 4 == False	*=	Multiply and assign	x = 1; x *= 2
>	Greater than	4 > 4 == False	/=	Divide and assign	x = 1; x /= 2
<=	Less than equal	4 <= 4 == True	//=	Floor divide and assign	x = 1; x //= 2
>=	Greater than equal	4 >= 4 == True	%=	Modulus assign	x = 1; x %= 2
==	Equal	4 == 5 == False	**=	Power assign	x = 1; x **= 2
!=	Not equal	4 != 5 == True	or, and, not	Boolean Or,	x = 1; x  = 2
<>	Not equal	4 <> 5 == True		Boolean And,	(a or b) and c
( )	Parenthesis	len("hi") == 2		Boolean Not	
[ ]	List brackets	[1,3,4]			
{ }	Dict curly braces	{'x': 5, 'y': 10}			

Fig 1.1

### Variables and Data types in Python:

In Python, variables are considered as storage placeholders for texts and numbers. Python is dynamically typed, such that there is no need to declare what the type of each variable is when it is declared or initialized.

### Python Programming Constructs:

The order in which the code is executed is referred to as a sequence. Selection is the process of determining which block of code will be run based on a set of

criteria. Repetition is a construct that determines whether parts of a program will be run several times based on certain criteria.

Talking about conditional statements, in python we can achieve this thing by using following keywords: if, elif and else. And scope of statement is decided through indentation.

Then we have Iterative statements in python i.e. Loops. They are primary of two types 'for loop' and 'while loop'.

Iterations and Looping

```
[ ] #for loop
fruits = ["apple", "banana", "cherry"]
for x in fruits:
    print(x)
#for loop does not require indexing

apple
banana
cherry
```

```
[50] #while loop
i = 1
while i < 6:
    print(i)
    i += 1

1
2
3
4
5
```

Continue statement is used to tell python to skip the rest of the statements in a current loop construct and continue with the next iteration of the code block. Break, on the other hand, is used to completely break out of the loop.

## Data Structures in Python:

- **List:** In Python, a list is one of the most basic and significant data structures. Each item is separated by a "," and is defined by enclosing square brackets "["]. Lists can be described as a collection of things with a positional value (index value) starting at 0 for each item (zero). It is changeable, which means that its contents can be altered. A List can be altered with the help of different methods.

Append()	Add an element to the end of the list
Extend()	Add all elements of a list to the another list
Insert()	Insert an item at the defined index
Remove()	Removes an item from the list
Pop()	Removes and returns an element at the given index
Clear()	Removes all items from the list
Index()	Returns the index of the first matched item
Count()	Returns the count of number of items passed as an argument
Sort()	Sort items in a list in ascending order
Reverse()	Reverse the order of items in the list
copy()	Returns a copy of the list

round()	Rounds off to the given number of digits and returns the floating point number
sum()	Sums up the numbers in the list
cmp()	This function returns 1, if first list is "greater" than second list
max()	return maximum element of given list
min()	return minimum element of given list
len()	Returns length of the list or size of the list
filter()	tests if each element of a list true or not returns a list of the results after applying the given function to each item of a given iterable
map()	This function can have any number of arguments but only one expression, which is evaluated and returned.
lambda()	



- **Tuple:** A Tuple can be defined as an immutable list. It can not be altered. It is defined by initializing elements in between parentheses “( )”. Once a tuple has been created, you can not add or alter elements in the tuple. It has only two methods: count() and index().
- **Sets:** A set contains an unordered collection of unique and immutable objects. All kinds of operations that are applicable to a set can be used for sets.
- **Dictionary:** It is a python data structure that is used to store data in key-value pairs. They are a set of attributes that have corresponding values. It is an unordered, indexed, and changeable form of data that is written within curly braces. We can perform numerous actions on a dictionary be it printing keys of the dictionary or all the values in the dictionary, adding a new element into the dictionary or removing any element from it.
- **Strings:** Strings can be defined as a list or an ordered chain of characters. We can perform various operations or manipulations on these strings.

#### ▼ Strings

```

✓ [49] word = "Hello-World"
0s      print(word.split("-"))
      print(word.replace("Hello", "Hi"))
      print(word[::-1])
      print(word.isalnum())

['Hello', 'World']
Hi-World
dlrow-olleH
False

```

### Slicing Functions:

The Python slice() function allows us to slice a sequence. It means we can retrieve a part of a string, tuple, list, etc. We can specify the start, end, and step of the slice. The step lets you skip items in the sequence.

These functions are very useful and I have used them in my project.

## Functions:

A function is a construct that is defined by keyword “def”.

```
[101] def add(a,b):  
      c = a+b  
      print(c)  
      add(2,3)  
  
5
```

Another type of function that is used a lot is Lambda Function. We use lambda functions when we require a nameless function for a short period of time.

We use Lambda Functions with Filter(), map() or reduce(). These functions are really very useful in data analysis.

## Classes and Objects

A class is a user-defined blueprint or prototype from which objects are created. Classes provide a means of bundling data and functionality together. Creating a new class creates a new type of object, allowing new objects of that type to be made. Each class instance can have attributes attached to it for maintaining its state.

\_\_init\_\_ method is used to initialize the attributes for a class with specific values for a particular object. It is executed at the time of object creation for a particular class.

```
[119] class Person:  
      # init method or constructor  
      def __init__(self, name):  
          self.name = name  
      # Sample Method  
      def say_hi(self):  
          print('Hello, my name is', self.name)  
p = Person('Robert')  
p.say_hi()  
  
Hello, my name is Robert
```

## NumPy

Numpy is a library for the python programming language adding support to large, multi-dimensional arrays and matrices along with a large collection of high-level mathematical functions to operate on these arrays.

We use `import numpy as np` to import numpy into our program. We use NumPy over lists because lists are slow to process. And NumPy aims to provide an array object that is up to 50x faster than traditional python list. The array object in Numpy is called `ndarray`. It supports a lot of supporting functions. Numpy is useful when it comes to array multiplication.

**Ndarray:** An ndarray is a multidimensional container of items of the same type and size.

**Ndarray.shape:** Shape is a tuple of integers representing the size of the ndarray in each dimension.

**Ndarray.size:** Size is the total number of elements in the ndarray. It is equal to the product of elements of the shape.

```
▶ #ndarray.size  
arr = np.array([[3,4,6], [0,8,1]])  
  
res = np.size(arr)  
print (res)
```

6

**Ndarray.dtype:** It tells the data type of the elements of a numpy array. In the numpy array all the elements have the same data type.

**Ndarray.itemsize:** Itemsize returns the size of each element of a numpy array.

## **Numpy Array indexing and slicing:**

### **Slicing**

Slicing in Python means taking elements from one given index to another given index. We pass slice instead of index like this - [start:end]

We can also define the step, like this: [start:end:step]

- If we don't mention start, it is set at 0 by default
- If we don't pass end , it is set at the length of array in that dimension
- If we don't mention pass, it is set at 1 by default.

### **Indexing**

“Indexing” means referring to an element of an iterable(for example, an array) by its position within the iterable.

### **Basic Functions:**

#### **Numpy.where()**

This function either replaces or processes the elements of the numpy array that satisfy the condition. **numpy.where(condition[,x,y])**.

Numpy.around() helps the user to evenly round array elements to the given number of decimals. It takes up to three parameters.

Numpy.floor() returns the floor value of the input array elements.

Numpy.ceil() returns smallest integer that is greater than or equal to that number.

### **Basic Statistical Functions:**

Numpy.sum() : It is used to find the sum of elements over a given axis. Default axis is None, will sum all of the elements of input array.

numpy.mean() function is used to compute the arithmetic mean along the specified axis. Returns the average of the array elements.

```
✓ 1s 1 #2D array
2 a = np.array([[1, 2], [3, 4]])
3 np.mean(a)

2.5
```

`numpy.median(arr, axis = None)` : Compute the median of the given data (array elements) along the specified axis.

```
✓ 0s [6] 1 # For 1D Array
2 arr = [2, 2, 7, 1, 3, 9, 34]
3
4 print("arr : ", arr)
5 print("Median of array : ", np.median(arr))
6

arr : [2, 2, 7, 1, 3, 9, 34]
Median of array : 3.0

✓ 0s [7] 1 # for 2D Array
2 arr=arr = [[14, 17, 12, 33, 44],
3           [15, 6, 27, 8, 19],
4           [23, 2, 54, 1, 4, ]]
5 print("\n Median of array, axis = 0 : ", np.median(arr, axis = 0))
6
7 # median along the axis = 1
8 print("\n Median of array, axis = 1 : ", np.median(arr, axis = 1))
9

Median of array, axis = 0 : [15.  6. 27.  8. 19.]
Median of array, axis = 1 : [17. 15.  4.]
```

`numpy.std(arr, axis = None)` : Compute the standard deviation of the given data (array elements) along the specified axis. The default, `axis=None`, will return the standard deviation of the given data. If `axis=0`, it will return the standard deviation along the column, and if `axis =1`, it will standard deviation along the rows.

## Pandas

### What is Pandas?

Pandas is a fast, powerful, flexible and easy to use open source data analysis and manipulation tool, built on top of the `_python` programming language. It takes data (like a CSV or TSV file, or a SQL database) and creates a Python object with rows and columns called data frame that looks very similar to a table in a statistical software (like Excel). After installing pandas onto our computer we need to import it into our program by: **Import pandas as pd**

## Data Structures in Pandas

### Series:

Pandas Series is a one-dimensional labeled array capable of holding any data type. So, in terms of Pandas DataStructure, A Series represents a single column in memory, which is either independent or belongs to a Pandas DataFrame.

### Dataframe:

Pandas DataFrame is two-dimensional size-mutable, potentially heterogeneous tabular data structure with labeled axes known as rows and columns. a dataframe is a collection of series that can be used to analyse the data. DataFrame can be created from the lists, dictionary, and from a list of dictionaries etc.

### Creating a series:

```
#import the pandas library and aliasing as pd
import pandas as pd
import numpy as np
data = np.array(['a','b','c','d'])
s = pd.Series(data)
print s
```

Its **output** is as follows –

```
0    a
1    b
2    c
3    d
dtype: object
```

Using Indexing:

```
#import the pandas library and aliasing as pd
import pandas as pd
import numpy as np
data = np.array(['a','b','c','d'])
s = pd.Series(data,index=[100,101,102,103])
print s
```

Its **output** is as follows –

```
100  a
101  b
102  c
103  d
dtype: object
```

We can read our .csv file in pandas by using `pd.read_csv('file_name', index_col =0)`. In this case `read_csv()` function returns a new dataframe with the data and the labels from the file which we have specified.

We even can create a series from a dictionary.

Accessing elements:

```
import pandas as pd
s = pd.Series([1,2,3,4,5],index = ['a','b','c','d','e'])

#retrieve the first three element
print s[:3]
```

Its **output** is as follows –

```
a  1
b  2
c  3
dtype: int64
```

## Sorting a DataFrame

To sort the DataFrame based on the values in a single column, use `.sort_values()`. By default, this will return a new DataFrame sorted in ascending order. It does not modify the original DataFrame.

**DataFrame.sort\_values(by, axis=0, ascending=True, inplace=False, kind='quicksort', na\_position='last', ignore\_index=False, key=None)**

*\* Axis: {0 or 'index', 1 or 'columns'}, default 0.*

## Null Handling

### isna()

The `isna()` function is used to check missing (null) values. It is a boolean function that looks for the missing values and returns TRUE where it detects a missing value.

Syntax: **dataframe.isna()**.

```
1 df.isna()
```

	Name	Age
0	False	False
1	False	False
2	False	True

With `notna()` function, we can easily pick out data that does not occupy missing values or NA values. The `notna()` function returns TRUE, if the data is free from missing values else it returns FALSE (if null values are encountered).

```
1 df.notna()
```

	Name	Age
0	True	True
1	True	True
2	True	False



## isnull()

isnull() function also works the same as isna(), detects missing values in the given series or dataframe. It returns a boolean same-sized object indicating if the values are NA (null). Missing values get mapped to True and non-missing values get mapped to False.

## fillna()

It manages and lets the user replace NaN values with some value of their own.

```
1 # replacing missing values in Age with 'Age missing'
2 dataf["Age"].fillna("Age missing", inplace = True)
3 dataf
```

	Name	Age
0	Adam	10.0
1	Steve	15.0
2	John	Age missing

## replace()

Pandas replace() is a very rich function that is used to replace a string, regex, dictionary, list, and series from the DataFrame. The values of the DataFrame can be replaced with other values dynamically. DataFrame.replace(to\_replace=" ", value=" ") .

## Aggregation of Groups (Group By)

Pandas group by is used for grouping the data according to the categories and apply a function to the categories. It also helps to aggregate data efficiently. Pandas dataframe.groupby() function is used to split the data into groups based on some criteria. DataFrame.groupby(by=None, axis=0, level=None, as\_index=True, sort=True, group\_keys=True, squeeze=False).

## Aggregate Functions

### Sum ()

groupby() function takes up the column name as argument followed by sum() function.

### Count ()

groupby() function takes up the column name as argument followed by count() function.

### Mean, Median, Mode

groupby() function takes up the column name as an argument followed by the name of the function or by using agg() function.

```
In [39]: 1 HouseOne.groupby("Subject")["Rating"].mean()
Out[39]: Subject
Subject_0    3.098475
Subject_1    2.389926
Subject_2    5.134058
Subject_3    3.381471
Subject_4    7.946482
Subject_5    5.380230
Name: Rating, dtype: float64

In [41]: 1 HouseOne.groupby("Subject")["Rating"].agg(np.mean)
Out[41]: Subject
Subject_0    3.098475
Subject_1    2.389926
Subject_2    5.134058
Subject_3    3.381471
Subject_4    7.946482
Subject_5    5.380230
Name: Rating, dtype: float64
```

### DESCRIBE function

The describe() method is used for calculating some statistical data like percentile, mean and std of the numerical values of the Series or DataFrame.

## Data Preprocessing

Data preprocessing is a process of preparing the raw data and making it suitable for a machine learning model. It is the first and crucial step while creating a machine learning model.

A real-world data generally contains noises, missing values, and maybe in an unusable format which cannot be directly used for machine learning models. Data preprocessing is required tasks for cleaning the data and making it suitable for a machine learning model which also increases the accuracy and efficiency of a machine learning model.

### **Constant Features:**

Constant features are those that show the same value, just one value, for all the observations of the dataset. This is the same value for all the rows of the dataset. These features provide no information that allows a machine learning model to discriminate or predict a target.

Identifying and removing constant features, is an easy first step towards feature selection and more easily interpretable machine learning models.

### **Quasi-constant Features:**

Quasi-constant features are those that show the same value for the great majority of the observations of the dataset. In general, these features provide little if any information that allows a machine learning model to discriminate or predict a target. But there can be exceptions. So we should be careful while removing these types of features.

We can use the `VarianceThreshold` function from `sklearn` library.

### **What is a Target Variable**

The target variable of a dataset is the feature of a dataset about which we want to gain a deeper understanding. A supervised machine learning algorithm uses historical data to learn patterns and uncover relationships between other features of your dataset and the target. The target variable will vary depending on the business goal and available data.

## Date Time

It is very important to convert the dates into friendly format and extract basic features of the data. Date variables are a special type of categorical variable. When pre-processing properly, they can highly enrich the dataset.

Series.dt can be used to access the values of the series of datetime variable and return several properties in the form of a numpy array.

*pandas.Series.dt.year returns the year of the date time.*

*pandas.Series.dt.month returns the month of the date time.*

*pandas.Series.dt.day returns the day of the date time.*

*pandas.Series.dt.quarter returns the quarter of the date time.*

If we want to extract month from date we can easily do that:

```
data['month'] = data['Date.of.Birth'].dt.month  
data[['Date.of.Birth','month']].head()
```

## Missing Values:

Missing data, or missing values, occur when no data / no value is stored for certain observations within a variable.

Incomplete data is an unavoidable problem in dealing with most data sources. Missing data is a common occurrence in both data science competitions and business datasets, and may have a significant impact on the conclusions that can be derived from the data.

There are several reasons why data can be missing:

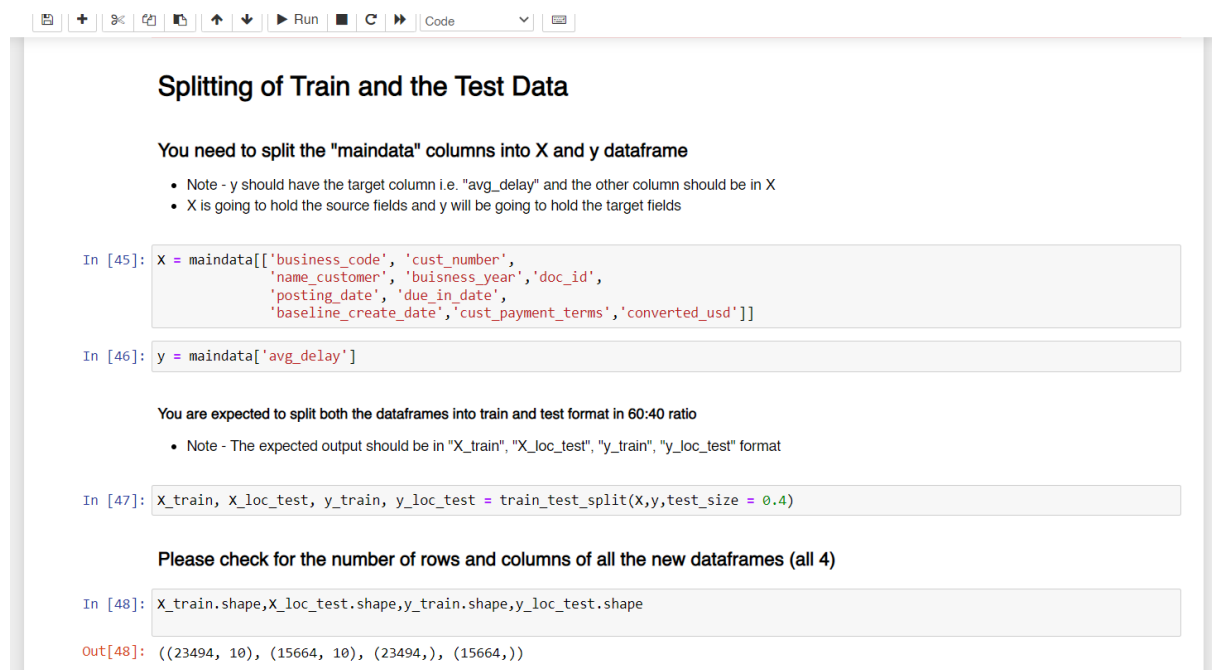
- A value is missing because it was forgotten, lost or not stored properly
- For a certain observation, the value of the variable does not exist
- The value can't be known or identified

## Train, Validation and Test

**Train Set:** The sample of data used to fit the model. The actual dataset that we use to train the model (weights and biases in the case of a Neural Network). The model sees and learns from this data.

**Validation Dataset:** The validation set is used to evaluate a given model, but this is for frequent evaluation. We use this data to fine-tune the model hyperparameters. Hence the model occasionally sees this data, but never does it “Learn” from this.

**Test dataset:** It provides the gold standard used to evaluate the model. It is only used once a model is completely trained (using the train and validation sets). The test set is generally what is used to evaluate competing model.



**Splitting of Train and the Test Data**

You need to split the "maindata" columns into X and y dataframe

- Note - y should have the target column i.e. "avg\_delay" and the other column should be in X
- X is going to hold the source fields and y will be going to hold the target fields

```
In [45]: x = maindata[['business_code', 'cust_number',
                    'name_customer', 'business_year', 'doc_id',
                    'posting_date', 'due_in_date',
                    'baseline_create_date', 'cust_payment_terms', 'converted_usd']]

In [46]: y = maindata['avg_delay']
```

You are expected to split both the dataframes into train and test format in 60:40 ratio

- Note - The expected output should be in "X\_train", "X\_loc\_test", "y\_train", "y\_loc\_test" format

```
In [47]: X_train, X_loc_test, y_train, y_loc_test = train_test_split(X,y,test_size = 0.4)
```

Please check for the number of rows and columns of all the new dataframes (all 4)

```
In [48]: X_train.shape,X_loc_test.shape,y_train.shape,y_loc_test.shape

Out[48]: ((23494, 10), (15664, 10), (23494,), (15664,))
```

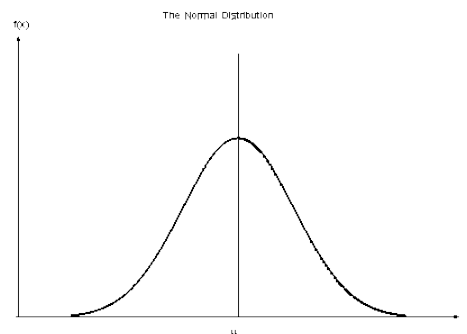
# Exploratory Data Analysis

## Continuous Variables

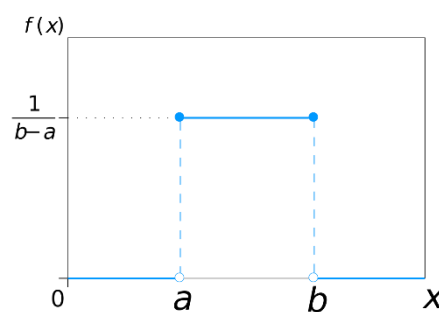
Continuous variables are numeric variables that have an infinite number of values between any two values. A Discrete variable is a variable whose value is obtained by counting.

A continuous distribution describes the probabilities of the possible values of a continuous random variable. A continuous random variable is a random variable with a set of possible values (known as the range) that is infinite and uncountable.

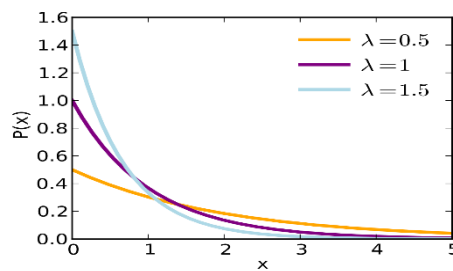
**Normal Distribution-** Normal distribution, also known as the Gaussian distribution, is a probability distribution that is symmetric about the mean, showing that data near the mean are more frequent in occurrence than data far from the mean. In graph form, normal distribution will appear as a bell curve.



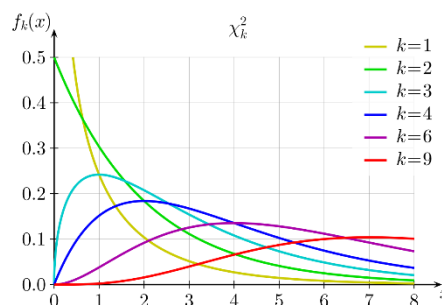
**Uniform Distribution** - A uniform distribution, sometimes also known as a rectangular distribution, is a distribution that has constant probability.



**Exponential Distribution** - In probability theory and statistics, the exponential distribution is the probability distribution of the time between events in a Poisson point process, i.e., a process in which events occur continuously and independently at a constant average rate. It is a particular case of the gamma distribution.



**Chi-Squared Distribution** - The Chi-Square distribution is the distribution of the sum of squared standard normal deviates. The degrees of freedom of the distribution is equal to the number of standard normal deviates being summed.



## Categorical Variable

A categorical variable is a variable that can take on one of a limited, and usually fixed number of possible values, assigning each individual or other units of observation to a particular group or nominal category on the basis of some qualitative property.

There are two types of Categorical Variables:

**Ordinal:** Categorical Variables which follow a certain order come under Ordinal Categorical Variables.

**Nominal:** Categorical Variables which don't follow any order come under nominal Categorical Variables.

## **Measure of Central Tendency: Mean, Median, Mode**

### **Mean**

The mean is the arithmetic average, and it is probably the measure of central tendency that you are most familiar. Add up all of the values and divide by the number of observations in dataset.

$$\frac{x_1 + x_2 + \dots + x_n}{n}$$

### **Median**

The median is the middle value. It is the value that splits the dataset in half. To find the median, order your data from smallest to largest, and then find the data point that has an equal amount of values above it and below it.

### **Mode**

The mode is the value that occurs the most frequently in your data set. On a bar chart, the mode is the highest bar.

Matplotlib- It is a basic plotting library

Seaborn- Most used plotting library

Plotly- New and interactive plotting library

## **Feature Engineering**



Feature engineering is the process of selecting, manipulating, and transforming raw data into features that can be used in supervised learning. In order to make machine learning work well on new tasks, it might be necessary to design and train better features.

**Feature Creation:** Creating features involves creating new variables which will be most helpful for our model. This can be adding or removing some features. As we saw above, the cost per sq. ft column was a feature creation.

**Transformations:** Feature transformation is simply a function that transforms features from one representation to another. The goal here is to plot and visualise data, if something is not adding up with the new features we can reduce the number of features used, speed up training, or increase the accuracy of a certain model.

**Feature Extraction:** Feature extraction is the process of extracting features from a data set to identify useful information. Without distorting the original relationships or significant information, this compresses the amount of data into manageable quantities for algorithms to process.

**Exploratory Data Analysis:** Exploratory data analysis (EDA) is a powerful and simple tool that can be used to improve your understanding of your data, by exploring its properties. The technique is often applied when the goal is to create new hypotheses or find patterns in the data.

**Benchmark:** A Benchmark Model is the most user-friendly, dependable, transparent, and interpretable model against which you can measure your own. It's a good idea to run test datasets to see if your new machine learning model outperforms a recognised benchmark.

**Feature scaling** is one of the most important data preprocessing step in machine learning. Algorithms that compute the distance between the features are biased towards numerically larger values if the data is not scaled.

**Normalization** or Min-Max Scaling is used to transform features to be on a similar scale. This scales the range to  $[0, 1]$  or sometimes  $[-1, 1]$ .

**Standardization** or Z-Score Normalization is the transformation of features by subtracting from mean and dividing by standard deviation. This is often called as Z-score.

## **Selecting Best Features**

Feature selection is the process of reducing the number of input variables when developing a predictive model.

### **Filter Method**

Filter methods are generally used as a preprocessing step. The selection of features is independent of any machine learning algorithms. Filter methods measure the relevance of features by their correlation with dependent variable.

### **Wrapper Method**

Wrapper methods measure the “usefulness” of features based on the classifier performance. wrapper methods are essentially solving the “real” problem (optimizing the classifier performance), but they are also computationally more expensive compared to filter methods due to the repeated learning steps and cross-validation.

### **Embedded Method**

Embedded methods combine the qualities’ of filter and wrapper methods. It’s implemented by algorithms that have their own built-in feature selection methods.



## Different Model Building Algorithms

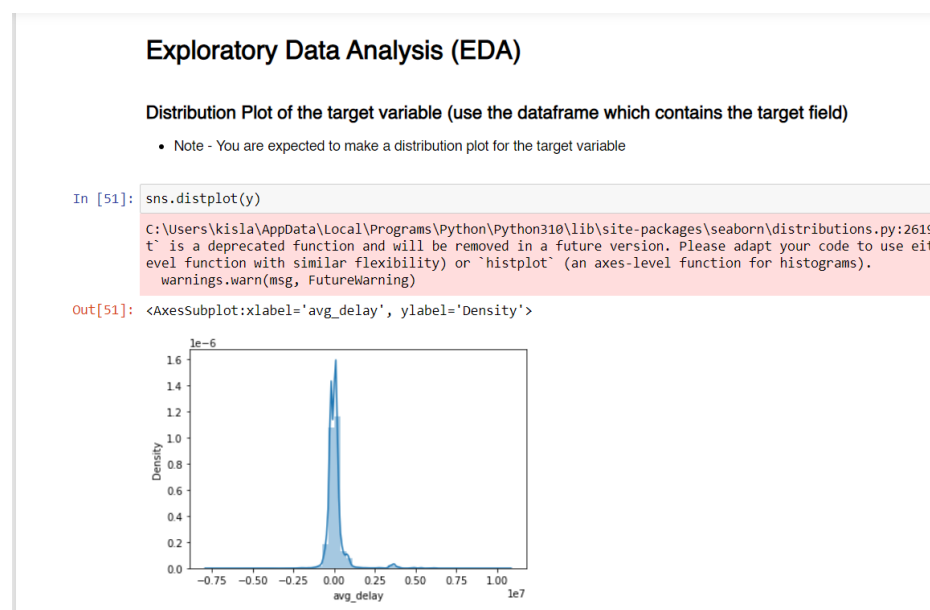
### Supervised learning

Supervised learning is an approach to creating artificial intelligence, where a computer algorithm is trained on input data that has been labeled for a particular output. The model is trained until it can detect the underlying patterns and relationships between the input data and the output labels, enabling it to yield accurate labeling results when presented with never-before-seen data. Supervised learning is good at classification and regression problems.

### Unsupervised learning

Unsupervised learning is a type of machine learning in which models are trained using unlabeled dataset and are allowed to act on that data without any supervision. Unsupervised learning cannot be directly applied to a regression or classification problem because unlike supervised learning, we have the input data but no corresponding output data.

In our project since we already had the output data therefore we went for regressor model and chose XGBoost model. Adding the screenshot for the code of feature engineering



You are expected to group the X\_train dataset on 'name\_customer' column with 'doc\_id' in the x\_train set

Need to store the outcome into a new dataframe

- Note code given for groupby statement- `X_train.groupby(by=['name_customer'], as_index=False)['doc_id'].count()`

```
In [52]: x_train_set = X_train.groupby(by=['name_customer'], as_index=False)['doc_id'].count()
x_train_set
```

```
Out[52]:
```

	name_customer	doc_id
0	99 CE	2
1	99 CE associates	1
2	99 CE co	1
3	99 CE corporation	1
4	99 CE foundation	2
...	...	...
3059	Y H trust	1
3060	YAEGER in	1
3061	YEN BROS	1
3062	YEN BROS co	1
3063	ZARCO co	1

3064 rows x 2 columns

## Feature Selection

### Filter Method

- Calling the VarianceThreshold Function
- Note - Keep the code as it is, no need to change

```
In [80]: from sklearn.feature_selection import VarianceThreshold
constant_filter = VarianceThreshold(threshold=0)
constant_filter.fit(X_train)
len(X_train.columns[constant_filter.get_support()])
```

```
Out[80]: 16
```

- Note - Keep the code as it is, no need to change

```
In [81]: constant_columns = [column for column in X_train.columns
                             if column not in X_train.columns[constant_filter.get_support()]]
print(len(constant_columns))
```

```
0
```

- transpose the feature matrice
- print the number of duplicated features
- select the duplicated features columns names
- Note - Keep the code as it is, no need to change

```
In [82]: x_train_T = X_train.T
```

#### Filtering depending upon correlation matrix value

- We have created a function called handling correlation which is going to return fields based on the correlation matrix value with a threshold of 0.8
- Note - Keep the code as it is, no need to change

```
In [83]: def handling_correlation(X_train,threshold=0.8):  
        corr_features = set()  
        corr_matrix = X_train.corr()  
        for i in range(len(corr_matrix.columns)):  
            for j in range(i):  
                if abs(corr_matrix.iloc[i, j]) > threshold:  
                    colname = corr_matrix.columns[i]  
                    corr_features.add(colname)  
        return list(corr_features)
```

- Note : Here we are trying to find out the relevant fields, from X\_train
- Please fill in the blanks to call handling\_correlation() function with a threshold value of 0.85

```
In [84]: train=X_train.copy()  
        handling_correlation(train.copy(),0.85)
```

```
Out[84]: ['month_of_createdate',  
         'month_of_due',  
         'year_of_due',  
         'year_of_postingdate',  
         'year_of_createdate',  
         'day_of_createdate']
```

Here is the Mean Square Error for the test dataset and for Validation dataset also the R2 score for the test and R2 score for the validation.

Calculate the Mean Square Error for test dataset

- Note - No need to change the code

```
In [111]: mean_squared_error(y_test,predictedfinal,squared=False)  
Out[111]: 376891.23648748884
```

#### Calculate the mean Square Error for validation dataset

```
In [112]: mean_squared_error(y_val,predictedfinal,squared=False)  
Out[112]: 912474.9153448767
```

#### Calculate the R2 score for test

```
In [113]: r2_score(y_test, predictedfinal)  
Out[113]: 0.688588781998222
```

#### Calculate the R2 score for Validation

```
In [114]: r2_score(y_val,predict_testfinal)  
Out[114]: 0.7108032947391175
```

#### Calculate the Accuracy for train Dataset

The screenshot shows a Jupyter Notebook interface with the following content:

```

Calculate the Accuracy for train Dataset

In [115]: regressor.score(X_train,y_train)
Out[115]: 0.9550187503072649

Calculate the accuracy for validation

In [116]: regressor.score(X_val,y_val)
Out[116]: 0.7108032947391175

Calculate the accuracy for test

In [117]: regressor.score(X_test,y_test)
Out[117]: 0.688588781998222

Specify the reason behind choosing your machine learning model

```

Each code cell includes a future warning from pandas regarding the deprecation of `Int64Index` in favor of `Index`.

Here is the screenshot for the accuracy of our model for train dataset and validation dataset and for the test. The accuracy for the train dataset was found to be around 95% , for the validation it was 71% and for the test it was found to be 68%.

The screenshot shows a Jupyter Notebook with the following content:

```

/59.11/0, 23/285929b/4.6183, 13/801450682.0580], [0.3273544554394432, -0.0036317721064091923, 0.479/9543235369637, 0.69/8963557944424],

The last but not the least model would be XGBoost or Extreme Gradient Boost Regression

• Step 1 : Call the XGBoost Regressor from xgb library
• Step 2 : make an object of Xgboost
• Step 3 : fit the X_train and y_train dataframe into the object
• Step 4 : Predict the output by passing the X_test Dataset into predict function

• Note - Append the Algorithm name into the algorithm list for tracking purpose### Extreme Gradient Boost Regression
• Note - No need to change the code

In [105]: import xgboost as xgb
Algorithm.append('XGB Regressor')
regressor = xgb.XGBRegressor()
regressor.fit(X_train, y_train)
predicted = regressor.predict(X_test)

C:\Users\kisla\AppData\Local\Programs\Python\Python310\lib\site-packages\xgboost\compat.py:36: FutureWarning: pandas.Int64Index
is deprecated and will be removed from pandas in a future version. Use pandas.Index with the appropriate dtype instead.
from pandas import MultiIndex, Int64Index
C:\Users\kisla\AppData\Local\Programs\Python\Python310\lib\site-packages\xgboost\data.py:262: FutureWarning: pandas.Int64Index
is deprecated and will be removed from pandas in a future version. Use pandas.Index with the appropriate dtype instead.
elif isinstance(data.columns, (pd.Int64Index, pd.RangeIndex)):

Check for the

• Mean Square Error
• R Square Error

for y_test and predicted dataset and store those data inside respective list for comparison

```

We had used Xtreme Gradient boost regressor model to train our dataset because It has the best combination of prediction performance and processing time compared to other algorithms.

## **Module 2: Web Application Development**

### **JAVA**

In this module we learnt about java and java basics and java servlets

#### **Object oriented Programming**

An object is a physical thing, such a pen, chair, table, computer, watch, etc. Any entity that has state and behaviour is know as an object. It is defined as an instance of a class.

**Class** is a collection of objects. It is a logical entity. A class can be defined as a blueprint from which we can create an individual object.

#### **Inheritance**

When one object acquires all the properties and behaviors of a parent object, it is known as inheritance. It provides code reusability. It is used to achieve runtime polymorphism.

In Java servlet we use this concept when we create a servlet class by extending servlet class.

#### **Polymorphism**

If one task is performed in different ways, it is known as polymorphism. In Java, we use method overloading and method overriding to achieve polymorphism.

#### **Abstraction**



Hiding internal details and showing functionality is known as abstraction. For example phone call, we don't know the internal processing.

## Encapsulation

Binding (or wrapping) code and data together into a single unit are known as encapsulation. For example, a capsule, it is wrapped with different medicines.

## Coupling

Coupling refers to the knowledge or information or dependency of another class. It arises when classes are aware of each other. If a class has the details information of another class, there is strong coupling.

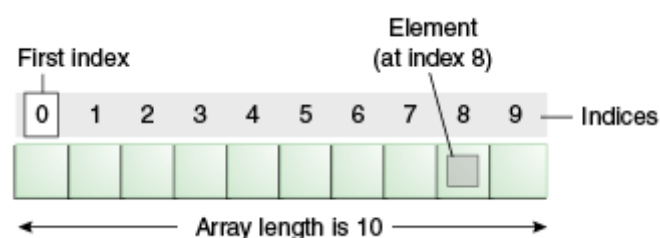
## Cohesion

Cohesion refers to the level of a component which performs a single well-defined task. A single well-defined task is done by a highly cohesive method. The weakly cohesive method will split the task into separate parts.

## Java Arrays

Array is a collection of similar types of elements which has contiguous memory location.

Java array is an object which contains elements of a similar data type. Array in Java is index-based, the first element of the array is stored at the 0th index, 2nd element is stored on 1st index and so on.



Syntax to declare array in java

1. `dataType[] arr;`
2. `dataType []arr;`
3. `dataType arr[];`

Java supports the feature of an anonymous array in which we don't need declare an array while passing the array to the method.

## **Java ArrayList**

The ArrayList class is a resizable [array](#), which can be found in the java.util package. The difference between a built-in array and an ArrayList in Java, is that the size of an array cannot be modified (if you want to add or remove elements to/from an array, you have to create a new one). While elements can be added and removed from an ArrayList .

```
ArrayList<String> cars = new ArrayList<String>();
```

The array list has many useful methods, to add elements to the array list we have add() method. If we want to access an element then we can use get() method in which we need to pass the index as an argument to the method to get the element. We have set() method to modify an element in array list in which we pass two arguments, first being the index number and other is the value. We have remove() method in array list which removes the element at particular index. Clear() method remove all elements of the array list. Size() method returns the size of the list.

## **Control Flow in Java**

### **If Statement:**

In Java, the "if" statement is used to evaluate a condition. The control of the program is diverted depending upon the specific condition. The condition of the If statement gives a Boolean value, either true or false. We have different types

of if statements: simple if, if-else statement, if-else-if ladder, Nested if-statement.

### **Switch Statement:**

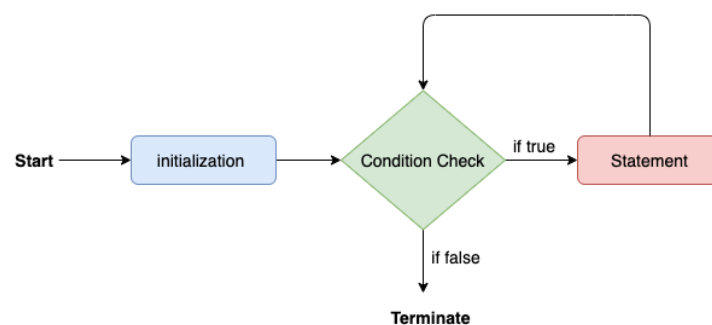
In Java, Switch statements are similar to if-else-if statements. The switch statement contains multiple blocks of code called cases and a single case is executed based on the variable which is being switched. Break statement terminates the switch block when the condition is satisfied. It is optional, if not used, next case is executed.

## **Loop Statements**

### **For Loop**

```
for(initialization, condition, increment/decrement) {  
    //block of statements  
}
```

It enables us to initialize the loop variable, check the condition, and increment/decrement in a single line of code.



### **Java for-each loop**

Java provides an enhanced for loop to traverse the data structures like array or collection. In the for-each loop, we don't need to update the loop variable.

```
for(data_type var : array_name/collection_name){  
    //statements
```

```
}
```

### Java while loop

The while loop is also used to iterate over the number of statements multiple times. It is also known as the entry-controlled loop since the condition is checked at the start of the loop. If the condition is true, then the loop body will be executed.

```
while(condition){  
//looping statements  
}
```

### Java do-while loop

The do-while loop checks the condition at the end of the loop after executing the loop statements. It is also known as the exit-controlled loop since the condition is not checked in advance.

```
do  
{  
//statements  
} while (condition);
```

## **Jump Statements**

Jump statements are used to transfer the control of the program to the specific statements.

### **Java break statement**

As the name suggests, the break statement is used to break the current flow of the program and transfer the control to the next statement outside a loop or switch statement.

## **Exception Handling in Java**

An exception is an event that disrupts the normal flow of the program. It is an object which is thrown at runtime.

**Exception Handling** is a mechanism to handle runtime errors such as `ClassNotFoundException`, `IOException`, `SQLException`, `RemoteException`, etc. The core advantage of exception handling is to maintain the normal flow of the application. An exception normally disrupts the normal flow of the application.

### **Types of Java Exceptions**

There are mainly two types of exceptions: checked and unchecked. An error is considered as the unchecked exception.

- Checked Exception
- Unchecked Exception
- Error

The classes that directly inherit the `Throwable` class except `RuntimeException` and `Error` are known as checked exceptions.

### **Unchecked Exception**

The classes that inherit the `RuntimeException` are known as unchecked exceptions. For example, `ArithmeticException`, `NullPointerException`, `ArrayIndexOutOfBoundsException`, etc.

### **Error**

Error is irrecoverable. Some example of errors are `OutOfMemoryError`, `VirtualMachineError`, `AssertionError` etc.

The "try" keyword is used to specify a block where we should place an exception code. The try block must be followed by either catch or finally.

The "catch" block is used to handle the exception. It must be preceded by try block which means we can't use catch block alone. It can be followed by finally block later.

The "finally" block is used to execute the necessary code of the program. It is executed whether an exception is handled or not.

The "throw" keyword is used to throw an exception.

The "throws" keyword is used to declare exceptions. It specifies that there may occur an exception in the method. It doesn't throw an exception. It is always used with method signature.

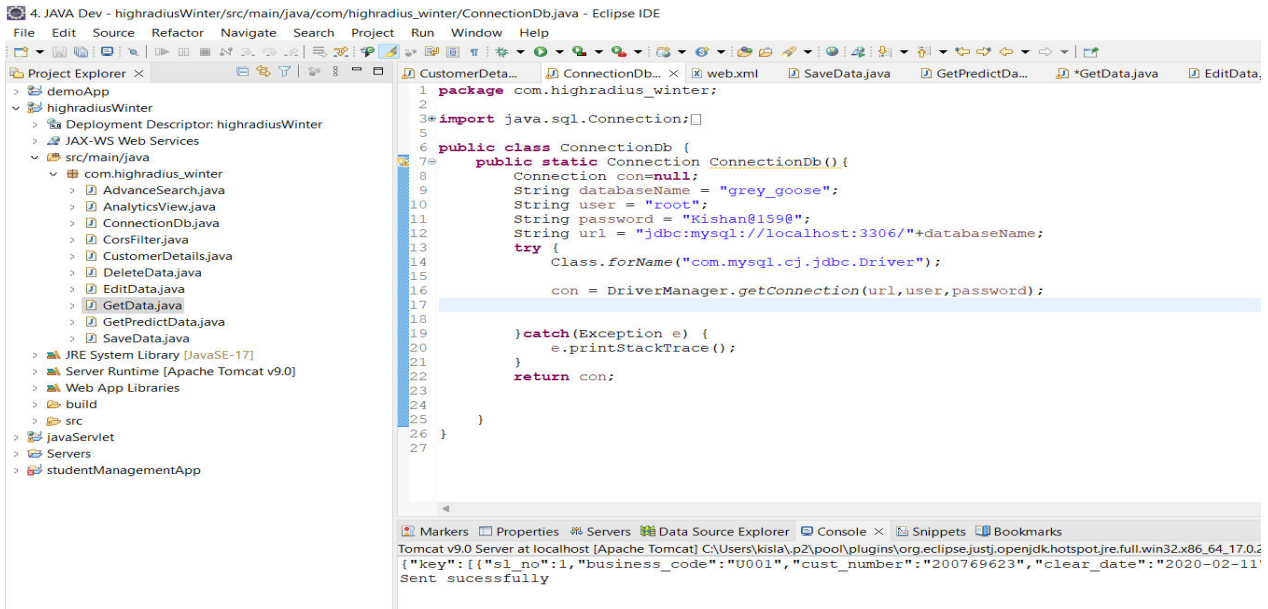
## **JDBC (Java Database Connectivity)**

JDBC stands for Java Database Connectivity. JDBC is a Java API to connect and execute the query with the database. It is a part of JavaSE (Java Standard Edition). JDBC API uses JDBC drivers to connect with the database. There are four types of JDBC drivers:

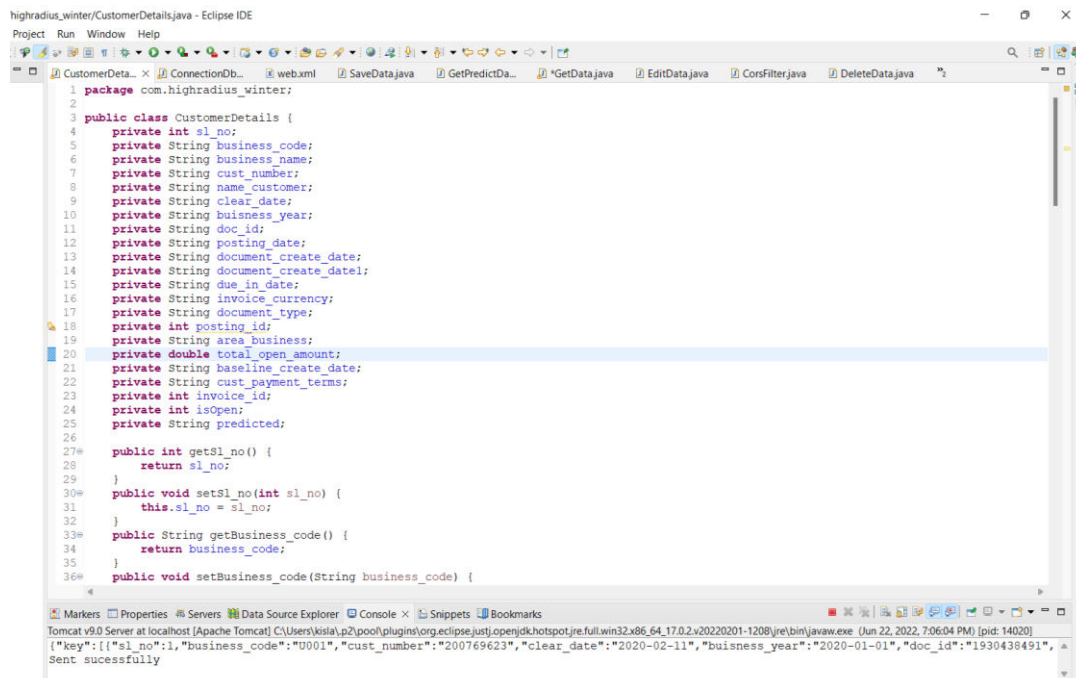
- JDBC-ODBC Bridge Driver,
- Native Driver,
- Network Protocol Driver, and
- Thin Driver

We can use JDBC API to handle database using Java program and can perform the following activities:

- Connect to the database
- Execute queries and update statements to the database
- Retrieve the result received from the database.



The JDBC Connection class, `java.sql.Connection`, represents a database connection to a relational database. Before reading or writing data from and to a database via JDBC, we need to open a connection to the database. Then only we can make changes to our database.



POJO class

```

1 package com.highradius_winter;
2
3 import java.io.IOException;
4
5 public class SaveData extends HttpServlet{
6
7     private static final long serialVersionUID = 1L;
8
9     protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
10
11         String jsonString = IOUtils.toString(request.getInputStream());
12
13         try {
14             JSONObject jsonObj = new JSONObject(jsonString.toString());
15
16             String business_code= jsonObj.getString("business_code");
17             int customer_number = jsonObj.getInt("customer_number");
18             String clear_date = jsonObj.getString("clear_date");
19             int business_year = Integer.parseInt( jsonObj.getString("business_year"));
20             String doc_id = jsonObj.getString("document_id");
21             String posting_date = jsonObj.getString("posting_date");
22             String document_create_date = jsonObj.getString("document_create_date");
23             String document_create_date1 = jsonObj.getString("document_create_date");
24             String due_date = jsonObj.getString("due_date");
25             String invoice_currency = jsonObj.getString("invoice_currency");
26             String doc_type = jsonObj.getString("doc_type");
27             int posting_id = Integer.parseInt( jsonObj.getString("posting_id"));
28             Double total_open_amount = Double.parseDouble( jsonObj.getString("total_open_amount"));
29             String baseline_create_date = jsonObj.getString("baseline_create_date");
30             String customer_payment_terms = jsonObj.getString("customer_payment_terms");
31             int invoice_id = Integer.parseInt( jsonObj.getString("invoice_id"));
32
33             Connection con;
34             con = ConnectionDb.ConnectionDb();
35
36         } catch (Exception e) {
37             e.printStackTrace();
38         }
39     }
40 }

```

Save data class

Get Data Class

```

81         arr.add(cd);
82
83     }
84
85     }
86     JSONArray jsonarr = new JSONArray();
87     for(int i=0;i<arr.size();i++) {
88         Gson gs = new Gson();
89         String jsonString = gs.toJson(arr.get(i));
90         jsonarr.add(jsonString);
91     }
92
93     JSONObject jsonObj = new JSONObject();
94     jsonObj.put("key", jsonarr);
95
96
97     System.out.println(jsonobj.toString());
98
99     String finalJson = jsonObj.toString().replace("\\\\\"", "\"").replace("\"{", "{").replace("}\"", "}");
100     System.out.println(finalJson);
101     PrintWriter out = response.getWriter();
102     response.setContentType("application/json");
103     response.setCharacterEncoding("utf-8");
104     out.write(finalJson);
105     System.out.println("Sent sucessfully");
106
107 } catch (SQLException e) {
108     e.printStackTrace();
109 }
110
111 }
112
113
114 }
115 }
116

```

Edit java class



```

}
}
} public class EditData extends HttpServlet {
    private static final long serialVersionUID = 1L;
    static Connection con;

    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        String jsonString = IOUtils.toString(request.getInputStream());
        try {
            JSONObject jsonObj = new JSONObject(jsonString.toString());
            String invoice_currency = jsonObj.getString("invoiceCurrency");
            String cust_payment_terms = jsonObj.getString("custPaymentTerms");
            int row_value = (int)jsonObj.getInt("row_value");
            con = ConnectionDb.ConnectionDb();
            PreparedStatement ps = con.prepareStatement("UPDATE winter_internship SET cust_payment_terms=?,invoice_currency=? WHERE sl_no=?");
            ps.setString(1, cust_payment_terms);
            ps.setString(2, invoice_currency);
            ps.setInt(3, row_value);

            int status = ps.executeUpdate();
            if(status!=0) {
                System.out.println("Updated successfully");
            }
        } catch (JSONException | SQLException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
    }
}
}
}

```

## Delete Data Class

```

CustomerData... ConnectionDb... web.xml SaveData.java GetPredictDa... *GetData.java EditData.java CorsFilter.java DeleteData.java x
2
3 *import java.io.IOException;
15
16 public class DeleteData extends HttpServlet{
17
18     protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
19         String jsonDeleteData = IOUtils.toString(request.getInputStream());
20         System.out.println(jsonDeleteData);
21
22         Connection con;
23         con = ConnectionDb.ConnectionDb();
24         JSONObject jsonObj;
25         try {
26             jsonObj = new JSONObject(jsonDeleteData.toString());
27
28             String invoice_currency = jsonObj.getString("data");
29             String modified_invoice_currency = invoice_currency.replace("[", "").replace("]", "");
30             String[] strArray = null;
31             strArray = modified_invoice_currency.split(",");
32             int[] intArray = new int[strArray.length];
33             for (int i = 0; i < strArray.length; i++){
34                 //
35                 System.out.println(strArray[i]);
36                 intArray[i] = Integer.parseInt(strArray[i]);
37             }
38             int status = 0;
39             for(int i:intArray) {
40                 System.out.println(i);
41                 PreparedStatement ps = con.prepareStatement("DELETE FROM testing_details WHERE sl_no=?");
42                 ps.setInt(1, i);
43                 status = ps.executeUpdate();
44             }
45             if (status!=0) {
46                 System.out.println("Deleted");
47             }
48         }
49     }
50 }

```

**CORS Filter** is a generic solution for fitting Cross-Origin Resource Sharing (CORS) support to Java web applications. CORS is a W3C standard for

enabling cross-domain requests from web browsers to servers and web APIs that opt in to handle them.

```
1 package com.highradius_winter;
2
3 import java.io.IOException;
4
5
6
7
8
9
10
11
12
13
14 public class CorsFilter extends HttpFilter {
15
16     /**
17      * @see Filter#destroy()
18      */
19     public void destroy() {
20         // TODO Auto-generated method stub
21     }
22
23     /**
24      * @see Filter#doFilter(ServletRequest, ServletResponse, FilterChain)
25      */
26     public void doFilter(ServletRequest request, ServletResponse response, FilterChain chain) throws IOException, ServletException {
27
28         HttpServletResponse res = (HttpServletResponse) response;
29
30         res.addHeader("Access-Control-Allow-Origin", "*");
31         res.addHeader("Access-Control-Allow-Headers", "*");
32         res.addHeader("Access-Control-Allow-Methods", "*");
33
34         chain.doFilter(request, response);
35     }
36
37     /**
38      * @see Filter#init(FilterConfig)
39      */
40     public void init(FilterConfig fConfig) throws ServletException {
41         // TODO Auto-generated method stub
42     }
43
44 }
```

Java web applications use a deployment descriptor file to determine how URLs map to servlets, which URLs require authentication, and other information.

```
44 <description></description>
45 <display-name>EditData</display-name>
46 <servlet-name>EditData</servlet-name>
47 <servlet-class>com.highradius_winter.EditData</servlet-class>
48 </servlet>
49 <servlet-mapping>
50 <servlet-name>EditData</servlet-name>
51 <url-pattern>/EditData</url-pattern>
52 </servlet-mapping>
53 <servlet>
54 <description></description>
55 <display-name>AdvanceSearch</display-name>
56 <servlet-name>AdvanceSearch</servlet-name>
57 <servlet-class>com.highradius_winter.AdvanceSearch</servlet-class>
58 </servlet>
59 <servlet-mapping>
60 <servlet-name>AdvanceSearch</servlet-name>
61 <url-pattern>/AdvanceSearch</url-pattern>
62 </servlet-mapping>
63 <servlet>
64 <description></description>
65 <display-name>AnalyticsView</display-name>
66 <servlet-name>AnalyticsView</servlet-name>
67 <servlet-class>com.highradius_winter.AnalyticsView</servlet-class>
68 </servlet>
69 <servlet-mapping>
70 <servlet-name>AnalyticsView</servlet-name>
71 <url-pattern>/AnalyticsView</url-pattern>
72 </servlet-mapping>
73 <servlet>
74 <description></description>
75 <display-name>DeleteData</display-name>
76 <servlet-name>DeleteData</servlet-name>
77 <servlet-class>com.highradius_winter.DeleteData</servlet-class>
78 </servlet>
```

web.xml defines mappings between URL paths and the servlets that handle requests with those paths.

## Front-end Development (HTML, CSS, ReactJS)

### HTML

The HyperText Markup Language or HTML is the standard markup language for documents designed to be displayed in a web browser. It can be assisted by technologies such as Cascading Style Sheets (CSS) and scripting languages such as JavaScript.

Web browsers receive HTML documents from a web server or from local storage and render the documents into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document.

HTML elements are the building blocks of HTML pages. With HTML constructs, images and other objects such as interactive forms may be embedded into the rendered page. HTML provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. HTML elements are delineated by tags, written using angle brackets. Tags such as `<img />` and `<input />` directly introduce content into the page. Other tags such as `<p>` surround and provide information about document text and may include other tags as sub-elements. Browsers do not display the HTML tags but use them to interpret the content of the page.

HTML can embed programs written in a scripting language such as JavaScript, which affects the behavior and content of web pages. Inclusion of CSS defines the look and layout of content. The World Wide Web Consortium (W3C), former maintainer of the HTML and current maintainer of the CSS standards, has encouraged the use of CSS over explicit presentational HTML since 1997.

A form of HTML, known as HTML5, is used to display video and audio, primarily using the **<canvas>** element, in collaboration with JavaScript.

```
<!DOCTYPE html>
<html>
  <head>
    <title>This is a title</title>
  </head>
  <body>
    <div>
      <p>Hello world!</p>
    </div>
  </body>
</html>
```

## CSS

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language such as HTML or XML (including XML dialects such as SVG, MathML or XHTML). CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript.

CSS is designed to enable the separation of presentation and content, including layout, colors, and fonts. This separation can improve content accessibility; provide more flexibility and control in the specification of presentation characteristics; enable multiple web pages to share formatting by specifying the relevant CSS in a separate **.css** file, which reduces complexity and repetition in the structural content; and enable the **.css** file to be cached to improve the page load speed between the pages that share the file and its formatting.

Separation of formatting and content also makes it feasible to present the same markup page in different styles for different rendering methods, such as on-screen, in print, by voice (via speech-based browser or screen reader), and on Braille-based tactile devices. CSS also has rules for alternate formatting if the content is accessed on a mobile device.

The name cascading comes from the specified priority scheme to determine which style rule applies if more than one rule matches a particular element. This cascading priority scheme is predictable.

The CSS specifications are maintained by the World Wide Web Consortium (W3C). Internet media type (MIME type) is registered for use with CSS by RFC 2318 (March 1998). The W3C operates a free CSS validation service for CSS documents.

### **Advantages of css:**

- **Create Stunning Web site** - CSS handles the look and feel part of a web page. Using CSS, you can control the color of the text, the style of fonts, the spacing between paragraphs, how columns are sized and laid out, what background images or colors are used, layout designs, variations in display for different devices and screen sizes as well as a variety of other effects.
- **Control web** - CSS is easy to learn and understand but it provides powerful control over the presentation of an HTML document. Most commonly, CSS is combined with the markup languages HTML or XHTML.
- **Pages load faster** - If you are using CSS, you do not need to write HTML tag attributes every time. Just write one CSS rule of a tag and apply it to all the occurrences of that tag. So less code means faster download times.

### **BASIC SYNTAX OF CSS**

```
body {  
  background-color: lightblue;  
}  
  
h1 {  
  color: white;  
  text-align: center;  
}  
  
p {  
  font-family: verdana;  
  font-size: 20px;  
}
```

## TYPE SELECTOR

```
h1 {  
  color: #36CFFF;  
}
```

## UNIVERSAL SELECTOR

```
* {  
  color: #000000;  
}
```

## CLASS SELECTOR

```
.black {  
  color: #000000;  
}
```

## ID SELECTOR

```
#black {  
  color: #000000;  
}
```

## CHILD SELECTOR

```
body > p {  
  color: #000000;  
}
```

## CSS USED IN THIS PROJECT ARE:

```
AddData.js  AddData.css X
D: > Highradius > client > src > Components > AddData.css
1  .AddData{
2      display: block;
3      background-color:#34495e;
4      position: absolute;
5      width: 90%;
6      height: 60%;
7      left: 50%;
8      top: 50%;
9      transform: translate(-50%,-50%);
10     z-index: 100;
11     padding-left: 30px;
12     padding-right: 30px;
13     border-radius: 10px;
14 }
15 .AddDataBox{
16     margin-top: 1%;
17     display: grid;
18     grid-template-columns: 1fr 1fr 1fr 1fr;
19     gap: 20px;
20 }
```

```
AddData.js  AddData.css X
D: > Highradius > client > src > Components > AddData.css > AddData
18     grid-template-columns: 1fr 1fr 1fr 1fr;
19     gap: 20px;
20 }
21 .addDataField{
22     position: relative;
23
24     width: 100%;
25     height: 7vh;
26     letter-spacing: 1px;
27     border-radius: 5px;
28     outline: 0;
29     border: 0;
30 }
31 .form_input{
32     height: 100%;
33     width: 100%;
34     outline: 0;
35     font-size: 1rem;
36     letter-spacing: 1px;
37
38 }
```

```
AddData.js  AddData.css X
D: > Highradius > client > src > Components > AddData.css > AddData
39 .form_label{
40     position: absolute;
41     left: 5px;
42     bottom: 0;
43     transition: 200ms ease-in;
44 }
45
46 .dateField {
47     position: absolute;
48     left: 5px;
49     font-size: 0.7rem;
50     top: 0;
51     transition: 200ms ease-in;
52 }
53 }
```

```
AddData.js  AddData.css X
D: > Highradius > client > src > Components > AddData.css > AddData
52
53 }
54 .form_input:focus ~ .form_label,
55 .form_input:not(:placeholder-shown).form_input:not(:focus) ~ .form_label{
56   top: 0;
57   font-size: 0.7rem;
58   font-weight: 500;
59 }
60 .AddData .button{
61   margin-top: 2%;
62   background: none;
63   border: 0;
64   width: 100%;
65   display: grid;
66   grid-template-columns: 1fr 1fr;
67   gap: 10px;
68
69
70 }
```

```
AddData.js  AddData.css X
D: > Highradius > client > src > Components > AddData.css > AddData
69
70 }
71 .AddData .button input{
72   background: none;
73   color: white;
74   outline: none;
75   cursor: pointer;
76   border: 2px solid white;
77   text-align: center;
78   font-size: 1rem;
79   border-radius: 5px;
80 }
81
82
< Run Testcases 0 0 tabnine starter See Tabnine Insights Ln 4, Col 13 Spaces: 4 UTF-8 CRLF CSS Go Live Prettier
```



## **LEARNING OUTCOMES**

During this internship training I got learn the fundamentals and basic of machine learning, web application development, JAVA, servlets, JDBC that definitely will help in my career.

Python, the foundational language for machine learning, served as our starting point. We began with Python, which is a foundational language for machine learning and the language we chose for our ML project. All of Python's modules were created and organized so that both fundamental and advanced concepts were covered.

I gained a foundation in probability as well as statistics for machine learning. Statistical Analysis is the discipline of the exploration of the collection of massive datasets to identify different hidden patterns and trends.

The most popular libraries for learning machine learning are numpy and pandas. Then got to study about other Machine learning libraries like sklearn, seaborn and matplotlib. Xtreme gradient boost regressor model was also introduced to me. Got to learn about data pre-processing and EDA also feature engineering. I

also had practical experience with training datasets and their division into train test and validation test.

I learned Java after the Machine Learning section because we were only told to code backend in Java for web application. Additionally, I learned how to create Java Servlet APIs for the Tomcat server. And learnt about JDBC to make a connection between servlet, frontend and database.

Learnt ReactJs a frontend library for building reusable UI components. It is an open source, component based front end library which is responsible for the view layer of the application. Learnt about props and hooks and how to integrate frontend and backend.

## **CONCLUSION**

The ground breaking technique known as machine learning is now widely discussed. Few people actually understand what machine learning is, despite the fact that the term has become popular. With the use of machine learning (ML), which is a form of artificial intelligence (AI), software programs can predict outcomes more accurately without having to be explicitly instructed to do so. In order to forecast new output values, machine learning algorithms use historical data as input. It also presents an uncertain route when it comes to making forecasts. React is one of the best frontend library for building reusable components.

Advantages: -

- Easily identifies trends and patterns
- No human intervention needed (automation)
- Machine Learning algorithms are good at handling data that are multi-dimensional and multi-variety

Disadvantages: -

- Data Privacy: Data is just like a fuel for many industries. Data utilized in the process to make decisions may breach the privacy of customers.
- Arbitrary Data May Yield Unexpected Results
- ML needs enough time to let the algorithms learn and develop enough to fulfill their purpose with a considerable amount of accuracy and relevancy.

## **References**

- <https://docs.python.org/3/tutorial/index.html>
- <https://reactjs.org/docs/getting-started.html>
- <https://numpy.org/doc/stable/user/whatisnumpy.html>
- <https://www.javatpoint.com/servlet-tutorial>
- <https://developer.mozilla.org/en-US/docs/Web/CSS>
- [https://pandas.pydata.org/pandas-docs/stable/user\\_guide/10min.html](https://pandas.pydata.org/pandas-docs/stable/user_guide/10min.html)
- <https://dev.mysql.com/doc/>

\*\*\*