KG COLLEGE OF ENGINEERING & TECHNOLOGY
(Approved by AICTE, New Delhi, Affiliated to JNTUH, Hyderabad)
Chilkur (Village), Moinabad (Mandal), R. R Dist, TS-501504

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING

MACHINE LEARNING

LAB MANUAL

Subject Code: CS601PC

Regulations: R 18 JNTUH

Class: III Year B. Tech. CSE I Semester/Semester II

COMPUTER SCIENCE AND ENGINEERING

KG REDDY COLLEGE OF ENGINEERING AND TECHNOLOGY

Affiliated to JNTUH, Chilkur,(V), Moinabad(M) R. R Dist, TS-501504

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

VISION AND MISSION OF THE INSTITUTION

VISION

To become self-sustainable institution which is recognized for its new age engineering through innovative teaching and learning culture, inculcating research and entrepreneurial ecosystem, and sustainable social impact in the community.

MISSION

- To offer undergraduate and post-graduate programs that is supported through industry relevant curriculum and innovative teaching and learning processes that would help students succeed in their professional careers.
- To provide necessary support structures for students, which will contribute to their personal and professional growth and enable them to become leaders in their respective fields.
- To provide faculty and students with an ecosystem that fosters research and development through strategic partnerships with government organisations and collaboration with industries.
- To contribute to the development of the region by using our technological expertise to work with nearby communities and support them in their social and economic growth.

VISION AND MISSION OF CSE DEPARTMENT

VISION

To be recognized as a department of excellence by stimulating a learning environment in which students and faculty will thrive and grow to achieve their professional, institutional and societal goals.

MISSION

- To provide high quality technical education to students that will enable life-long learning and build expertise in advanced technologies in Computer Science and Engineering.
- To promote research and development by providing opportunities to solve complex engineering problems in collaboration with industry and government agencies.
- To encourage professional development of students that will inculcate ethical values and leadership skills while working with the community to address societal issues

ProgramEducationalObjectives(PEOs):

AgraduateoftheComputerScience andEngineeringProgramshould:

ProgramEducationalObjective1:(PEO1)

The Graduates will provide solutions to difficult and challenging issues in their profession by applying computer science and engineering theory and principles.

ProgramEducationalObjective2:(PEO2)

The Graduates have successful careers in computer science and engineering fields or will be able to successfully pursue advanced degrees.

ProgramEducationalObjective3:(PEO3)

The Graduates will communicate effectively, work collaboratively and exhibit high levels of Professionalism, moral and ethical responsibility.

ProgramEducationalObjective4:(PEO4)

The Graduates will develop the ability to understand and analyse Engineering issues in a broader perspective with ethical responsibility towards sustainable development.

ProgramOutcomes(POs):

PO1	**Engineeringknowledge**:Applytheknowledgeofmathematics,science,engineering Fundamentals,andanengineeringspecializationtothesolutionofcomplexengineeringproblems.
PO2	**Problemanalysis**:Identify,formulate,reviewresearchliterature,andanalyzecomplexengineering problems reaching substantiated conclusions using first principles ofmathematics,natural sciences, andengineeringsciences.
PO3	**Design/developmentofsolutions**:Designsolutionsforcomplexengineeringproblemsanddesign system components or processes that meet the specified needs with appropriateconsideration for the public health and safety, and the cultural, societal, and environmentalconsiderations.
PO4	**Conduct investigations of complex problems**: Use research-based knowledge and researchmethods including design of experiments, analysis and interpretation of data, and synthesis oftheinformation to providevalid conclusions.
PO5	**Modern tool usage**: Create, select, and apply appropriate techniques, resources, and modernengineering and IT tools including prediction and modeling to complex engineering activitieswithan understandingofthelimitations.
PO6	**The engineer and society**: Apply reasoning informed by the contextual knowledge to assesssocietal, health, safety, legal and cultural issues and the consequent responsibilities relevant totheprofessional engineeringpractice.

PO7	**Environmentand sustainability**:Understandtheimpactoftheprofessionalengineering solutionsinsocietalandenvironmentalcontexts,anddemonstratetheknowledgeof,andneedforsustainabledevelopment.
PO8	**Ethics**: Apply ethical principles and commit to professional ethics and responsibilities andnormsoftheengineering practice.
PO9	**Individualand teamwork**:Functioneffectivelyasanindividual,andasa memberorleader indiverseteams,andinmultidisciplinarysettings.
PO10	**Communication**: Communicate effectively on complex engineering activities with theengineering community and with society at large, such as, being able to comprehend andwriteeffectivereportsanddesigndocumentation,makeeffectivepresentations,andgiveandreceiveclearinstructions.
PO11	**Projectmanagementand finance**:Demonstrateknowledgeandunderstandingofthe engineeringandmanagementprinciplesandapplythesetoone'sownwork,asamember andleaderinateam, to manageprojects andin multidisciplinaryenvironments.
PO12	**Life-long learning**: Recognize the need for, and have the preparation and ability to engage inindependentandlife-longlearningin thebroadest context oftechnological change.

ProgramSpecificOutcomes(PSOs):

PSO1	**Problem Solving Skills** – Graduate will be able to apply computational techniques and software principles to solve complex engineering problems pertaining to software engineering.
PSO2	**Professional Skills** – Graduate will be able to think critically, communicate effectively, and collaborate in teams through participation in co and extra-curricular activities.
PSO3	**Successful Career** – Graduates will possess a solid foundation in computer science and engineering that will enable them to grow in their profession and pursue lifelong learning through post-graduation and professional development.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

MACHINE LEARNING LAB

LIST OF EXPERIMENTS

S. No.	Name of the Experiment
1.	The probability that it is Friday and that a student is absent is 3 %. Since there are 5 school days in a week, the probability that it is Friday is 20 %. What is theprobability that a student is absent given that today is Friday? Apply Baye's rule in python to get the result. (Ans: 15%)
2.	Extract the data from database using python
3.	Implement k-nearest neighbours classification using python
4.	Given the following data, which specify classifications for nine ombinations of VAR1 and VAR2 predict a classification for a case where VAR1=0.906 and VAR2=0.606, using the result of k-means clustering with 3 means (i.e., 3 centroids) VAR1 VAR2 CLASS 1.713 1.586 0 0.180 1.786 1 0.353 1.240 1 0.940 1.566 0 1.486 0.759 1 1.266 1.106 0 1.540 0.419 1 0.459 1.799 1 0.773 0.186 1
5.	**The following training examples map descriptions of individuals onto high, medium and low**

	credit-worthiness. medium skiing design single twenties no ->highRisk high golf trading married forties yes ->lowRisk ow speedway transport married thirties yes ->medRisk medium football banking single thirties yes ->lowRisk high flying media married fifties yes ->highRisk ow football security single twenties no ->medRisk medium golf media single thirties yes ->medRisk medium golf transport married forties yes ->lowRisk high skiing banking single thirties yes ->highRisk ow golf unemployed married forties yes ->highRisk Input attributes are (from left to right) income, recreation, job, status, age-group, home-owner. Find the unconditional probability of `golf' and the conditional probability of `single' given `medRisk' in the dataset?
6.	Implement linear regression using python.
7.	Implement Naïve Bayes theorem to classify the English text
8.	Implement an algorithm to demonstrate the significance of genetic algorithm

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

MACHINE LEARNING LAB

INTRODUCTION TO LAB:

Machine Learning is used anywhere from automating mundane tasks to offering intelligent insights, industries in every sector try to benefit from it. You may already be using a device that utilizes it. For example, a wearable fitness tracker like Fitbit, or an intelligent home assistant like Google Home. But there are much more examples of ML in use.

- **Prediction:**Machine learning can also be used in the prediction systems. Considering the loan example, to compute the probability of a fault, the system will need to classify the available data in groups.
- **Image recognition**:Machine learning can be used for face detection in an image as well. There is a separate category for each person in a database of several people.
- **Speech Recognition:**It is the translation of spoken words into the text. It is used in voice searches and more. Voice user interfaces include voice dialing, call routing, and appliance control. It can also be used a simple data entry and the preparation of structured documents.
- **Medical diagnoses:**ML is trained to recognize cancerous tissues.
- **Financial industry:**andtrading:companies use ML in fraud investigations and credit checks.

Types of Machine Learning?

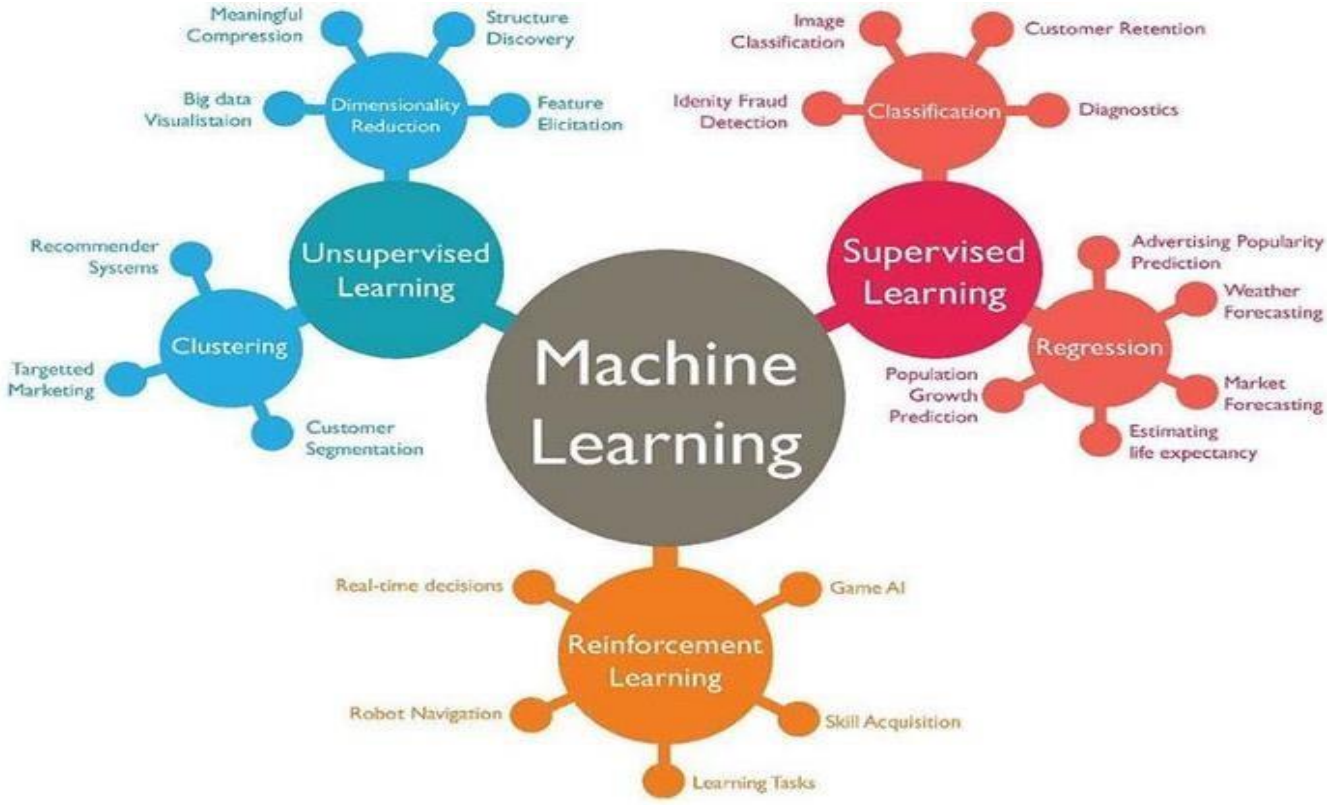
Machine learning can be classified into 3 types of algorithms

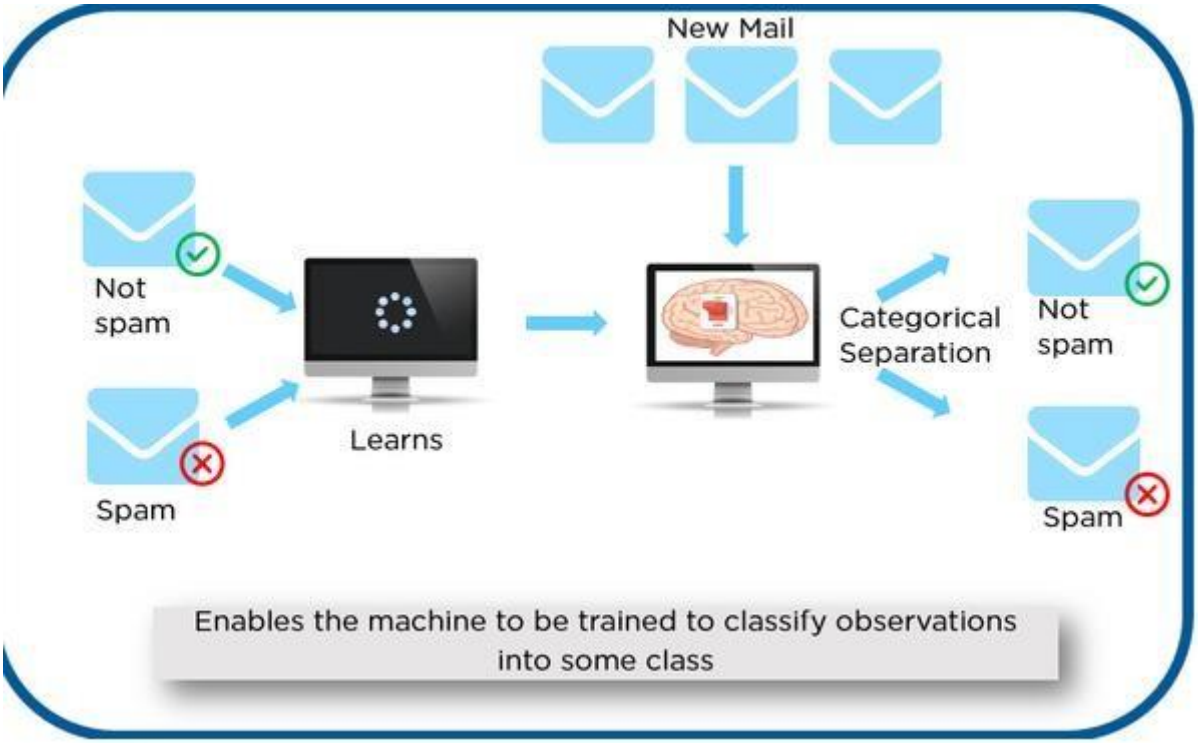
1. Supervised Learning
2. Unsupervised Learning
3. Reinforcement Learning

Overview of Supervised Learning Algorithm

In Supervised learning, an AI system is presented with data which is labeled, which means that each data tagged with the correct label.

The goal is to approximate the mapping function so well that when you have new input data (x) that you can predict the output variables (Y) for that data.





As shown in the above example, we have initially taken some data and marked them as 'Spam' or 'Not Spam'. This labeled data is used by the training supervised model, this data is used to train the model.

Once it is trained we can test our model by testing it with some test new mails and checking of the model is able to predict the right output.

Types of Supervised learning

- **Classification**: A classification problem is when the output variable is a category, such as "red" or "blue" or "disease" and "no disease".
- **Regression**: A regression problem is when the output variable is a real value, such as "dollars" or "weight".

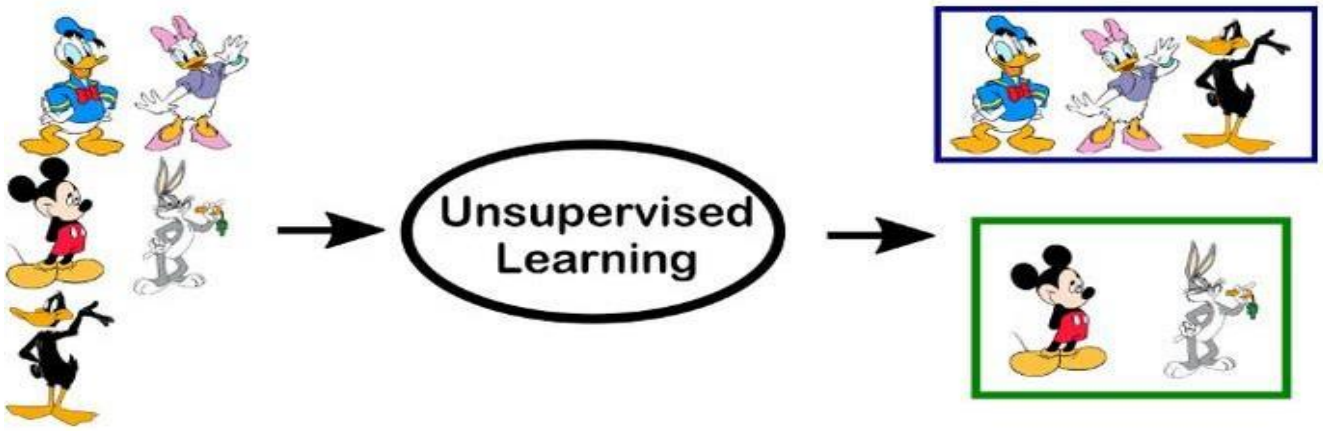
Overview of Unsupervised Learning Algorithm

In unsupervised learning, an AI system is presented with unlabeled, uncategorized data and the system's algorithms act on the data without prior training. The output is dependent upon the coded algorithms. Subjecting a system to unsupervised learning is one way of testing AI.

Types of Unsupervised learning:

- **Clustering**: A clustering problem is where you want to discover the inherent groupings in the data, such as grouping customers by purchasing behavior.

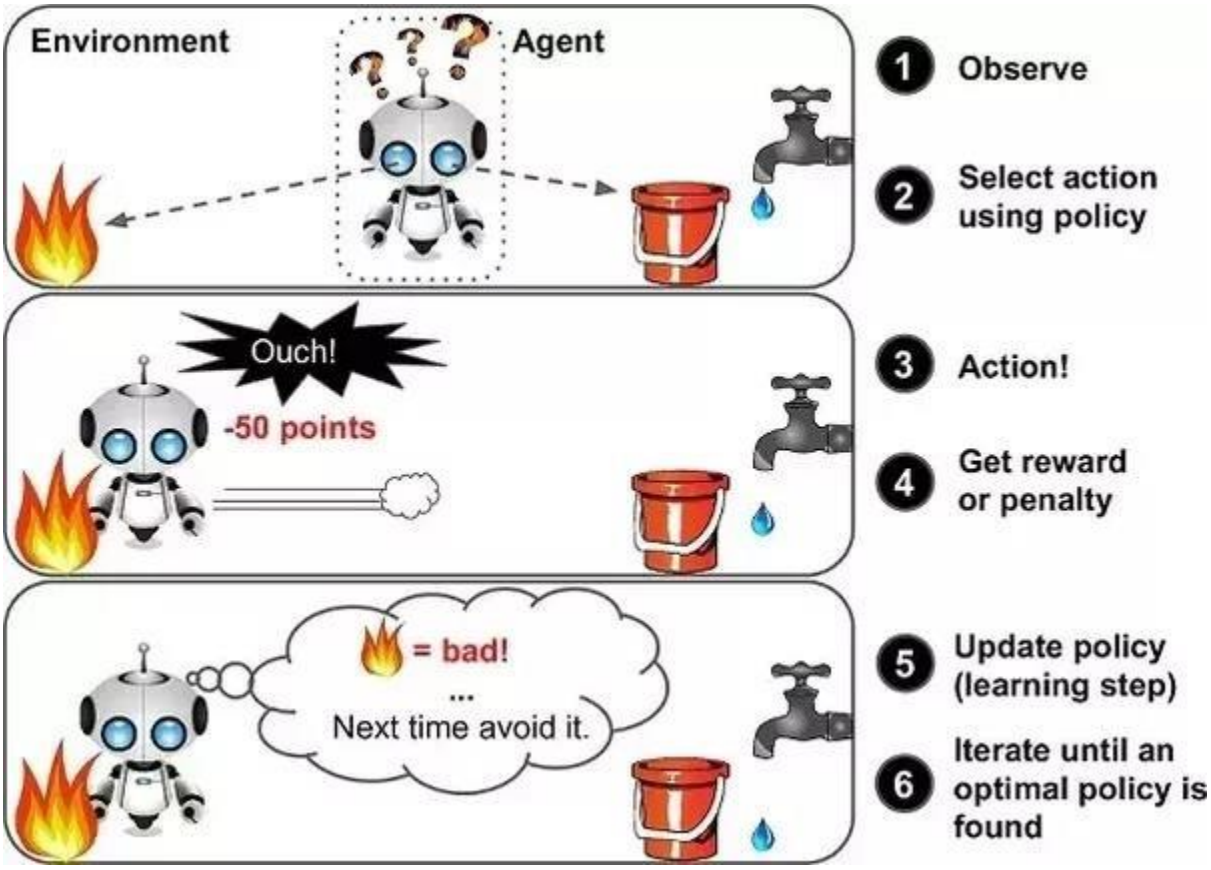
- **Association**: An association rule learning problem is where you want to discover rules that describe large portions of your data, such as people that buy X also tend to buy Y.



Example of Unsupervised Learning

Overview of Reinforcement Learning

A reinforcement learning algorithm, or agent, learns by interacting with its environment. The agent receives rewards by performing correctly and penalties for performing incorrectly. The agent learns without intervention from a human by maximizing its reward and minimizing its penalty. It is a type of dynamic programming that trains algorithms using a system of reward and punishment.



in the above example, we can see that the agent is given 2 options i.e. a path with water or a path with fire. A reinforcement algorithm works on reward a system i.e. if the agent uses the fire path then the rewards are subtracted and agent tries to learn that it should avoid the fire path. If it had chosen the water path or the safe path then some points would have been added to the reward points, the agent then would try to learn what path is safe and what path isn't.

It is basically leveraging the rewards obtained; the agent improves its environment knowledge to select the next action.

PROGRAM 1

The probability thatitisFridayandthatastudentisabsentis3%.Sincethereare5school daysinaweek,theprobabilitythatitisFridayis20%.Whatisthe probabilitythatastudent is absent given that today is Friday? Apply Baye's rule in python to get the result. (Ans: 15%)

AIM: To find the probability that a student is absent given that today is Friday.

DESCRIPTION:

Machine learning is a method of data analysis that automates analytical model building of data set. Using the implemented algorithms that iteratively learn from data, machine learning allows computers to find hidden insights without being explicitly programmed where to look. Naive bayes algorithm is one of the most popular machines learning technique. In this article we will look how to implement Naive bayes algorithm using python.

Before someone can understand Bayes' theorem, they need to know a couple of related concepts first, namely, the idea of Conditional Probability, and Bayes' Rule.

Conditional Probability is just what is the probability that something will happen, given that something else has already happened.

Let say we have a collection of people. Some of them are singers. They are either male or female. If we select a random sample, what is the probability that this person is a male? what is the probability that this person is a male and singer? Conditional Probability is the best option here. We can calculate probability like,

P(Singer & Male) = P(Male) x P(Singer / Male)

What is Bayes rule ?

We can simply define Bayes rule like this. Let A1, A2, …, An be a set of mutually exclusive events that together form the sample space S. Let B be any event from the same sample space, such that P(B) > 0. Then, P(Ak | B) = P(Ak ∩ B) / P(A1 ∩ B) + P(A2 ∩ B) + . . . + P(An ∩ B)

What is Bayes classifier?

Naive Bayes classifiers are a family of simple probabilistic classifiers based on applying Bayes' theorem with strong (naive) independence assumptions between the features in machine learning. Basically we can use above theories and equations for classification problem.

SOURCE CODE:

```
probAbsentFriday=0.0

3 probFriday=0.2

# bayes Formula

#p(Absent|Friday)=p(Friday|Absent)p(Absent)/p(Friday)

#p(Friday|Absent)=p(Friday∩Absent)/p(Absent)
# Therefore the result is:
bayesResult=(probAbsentFriday/probFriday)
print(bayesResult * 100)
```

Output: 15

```
probAbsentFriday=0.03
probFriday=0.2
# bayes Formula
#p(Absent|Friday)=p(Friday|Absent)p(Absent)/p(Friday)
#p(Friday|Absent)=p(FridaynAbsent)/p(Absent)
# Therefore the result is:
bayesResult=(probAbsentFriday/probFriday)
print(bayesResult * 100)
```

```
15.0
```

1.5.VIVA QUESTIONS & ANSWERS

1. What are Bayesian Networks (BN) ?

Bayesian Network is used to represent the graphical model for probability relationship among a set of variables. Bayes' theorem is a way to figure out conditional probability. Conditional probability is the probability of an event happening, given that it has some relationship to one or more other events. For example, your probability of getting a parking space is connected to the time of day you park, where you park, and what conventions are going on at any time. Bayes' theorem is slightly more nuanced. In a nutshell, it gives you the actual probability of an **event** given information about **tests**.

- "Events" Are different from "tests." For example, there is a **test** for liver disease, but that's separate from the **event** of actually having liver disease.
- **Tests are flawed**: just because you have a positive test does not mean you actually have the disease. Many tests have a high false positive rate. **Rare events tend to have higher false positive rates** than more common events. We're not just talking about medical tests here. For example, spam filtering can have high false positive rates. Bayes' theorem takes the test results and calculates your *real probability* that the test has identified the event.

2. Can you give any real time example using Bayes' Theorem (liver disease).

You might be interested in finding out a patient's probability of having liver disease if they are an alcoholic. "Being an alcoholic" is the **test** (kind of like a litmus test) for liver disease.

- **A** could mean the event "Patient has liver disease." Past data tells you that 10% of patients entering your clinic have liver disease. P(A) = 0.10.
- **B** could mean the litmus test that "Patient is an alcoholic." Five percent of the clinic's patients are alcoholics. P(B) = 0.05.
- You might also know that among those patients diagnosed with liver disease, 7% are alcoholics. This is your **B|A:** the probability that a patient is alcoholic, given that they have liver disease, is 7%.

Bayes' theorem tells you:

P(A|B)=(0.07*0.1)/0.05=0.14

In other words, if the patient is an alcoholic, their chances of having liver disease is 0.14 (14%).

This is a large increase from the 10% suggested by past data. But it's still unlikely that any particular patient has liver disease.

3. **Bayes' Theorem Examples 2:** ***what is the probability that they will be prescribed pain pills?***

Another way to look at the theorem is to say that one event follows another. Above I said "tests" and "events", but it's also legitimate to think of it as the "first event" that leads to the "second event." There's no one right way to do this: use the terminology that makes most sense to you.

In a particular pain clinic, 10% of patients are prescribed narcotic pain killers. Overall, five percent of the clinic's patients are addicted to narcotics (including pain killers and illegal substances). Out of all the people prescribed pain pills, 8% are addicts. *If a patient is an addict, what is the probability that they will be prescribed pain pills?*

Step 1:Figure out what your event "A" is from the question. That information is in the italicized part of this particular question. The event that happens first (A) is being prescribed pain pills. That's given as 10%.

Step 2:Figure out what your event "B" is from the question. That information is also in the italicized part of this particular question. Event B is being an addict. That's given as 5%.

Step 3:Figure out what the probability of event B (Step 2) given event A (Step 1). In other words, find what (B|A) is. We want to know "Given that people are prescribed pain pills, what's the probability they are an addict?" That is given in the question as 8%, or .8.

Step 4:Insert your answers from Steps 1, 2 and 3 into the formula and solve. P(A|B) = P(B|A) * P(A) / P(B) = (0.08 * 0.1)/0.05 = 0.16

The probability of an addict being prescribed pain pills is 0.16 (16%).

4. Bayes' Theorem Examples 3: the Medical Test if a person gets a positive test result.

what are the odds they actually have the genetic defect?

A slightly more complicated example involves a medical test (in this case, a genetic test):

There are **several forms of Bayes' Theorem** out there, and they are all equivalent (they are just written in slightly different ways). In this next equation, "X" is used in place of "B." In addition, you'll see some changes in the denominator. The proof of why we can rearrange the equation like this is beyond the scope of this article (otherwise it would be 5,000 words instead of 2,000!). However, if you come across a question involving medical tests, you'll likely be using this alternative formula to find the answer:

$$\Pr(\mathrm{A}|\mathrm{X}) = \frac{\Pr(\mathrm{X}|\mathrm{A})\Pr(\mathrm{A})}{\Pr(\mathrm{X}|\mathrm{A})\Pr(A) + \Pr(\mathrm{X}|\sim \mathrm{A})\Pr(\sim A)}$$

1% of people have a certain genetic defect.
90% of tests for the gene detect the defect (true positives).
9.6% of the tests are false positives.
If a person gets a positive test result, **what are the odds they actually have the genetic defect?**

The first step into solving Bayes' theorem problems is to assign letters to events:

- A = chance of having the faulty gene. That was given in the question as 1%. That also means the probability of *not* having the gene (~A) is 99%.
- X = A positive test result.

So:

1. P(A|X) = Probability of having the gene given a positive test result.
2. P(X|A) = Chance of a positive test result given that the person actually has the gene. That was given in the question as 90%.
3. p(X|~A) = Chance of a positive test if the person *doesn't* have the gene. That was given in the question as 9.6%

Now we have all of the information we need to put into the equation:
P(A|X) = (.9 * .01) / (.9 * .01 + .096 * .99) = 0.0865 (8.65%).

The probability of having the faulty gene on the test is 8.65%.

5.Given the following statistics, what is the probability that a woman has cancer if she has a positive mammogram result?

- One percent of women over 50 have breast cancer.
- Ninety percent of women who have breast cancer test positive on mammograms.
- Eight percent of women will have false positives.

Step 1: Assign events to A or X. You want to know what a woman's probability of having cancer is, given a positive mammogram. For this problem, actually having cancer is A and a positive test result is X.
Step 2: List out the parts of the equation (this makes it easier to work the actual equation):
P(A)=0.01
P(~A)=0.99
P(X|A)=0.9
P(X|~A)=0.08
Step 3: Insert the parts into the equation and solve. Note that as this is a medical test, we're using the form of the equation from example #2:
(0.9 * 0.01) / ((0.9 * 0.01) + (0.08 * 0.99) = 0.10.

The probability of a woman having cancer, given a positive test result, is 10%.

PROGRAM 2

2.EXTRACT THE DATA FROM DATABASE USING PYTHON

You'll learn the following MySQL SELECT operations from Python using a 'MySQL Connector Python' module.

- Execute the SELECT query and process the result set returned by the queryin Python.
- Use Python **variables in a where clause** of a SELECT query to pass dynamic values.
- Use fetchall(), fetchmany(), and fetchone() methods of a cursor class to fetch all or limited rows from atable.
- Python Select from MySQL Table

This article demonstrates how to **select rows of a MySQL table in Python**.

You'll learn the following MySQL SELECT operations from Python using a 'MySQL Connector Python' module

.

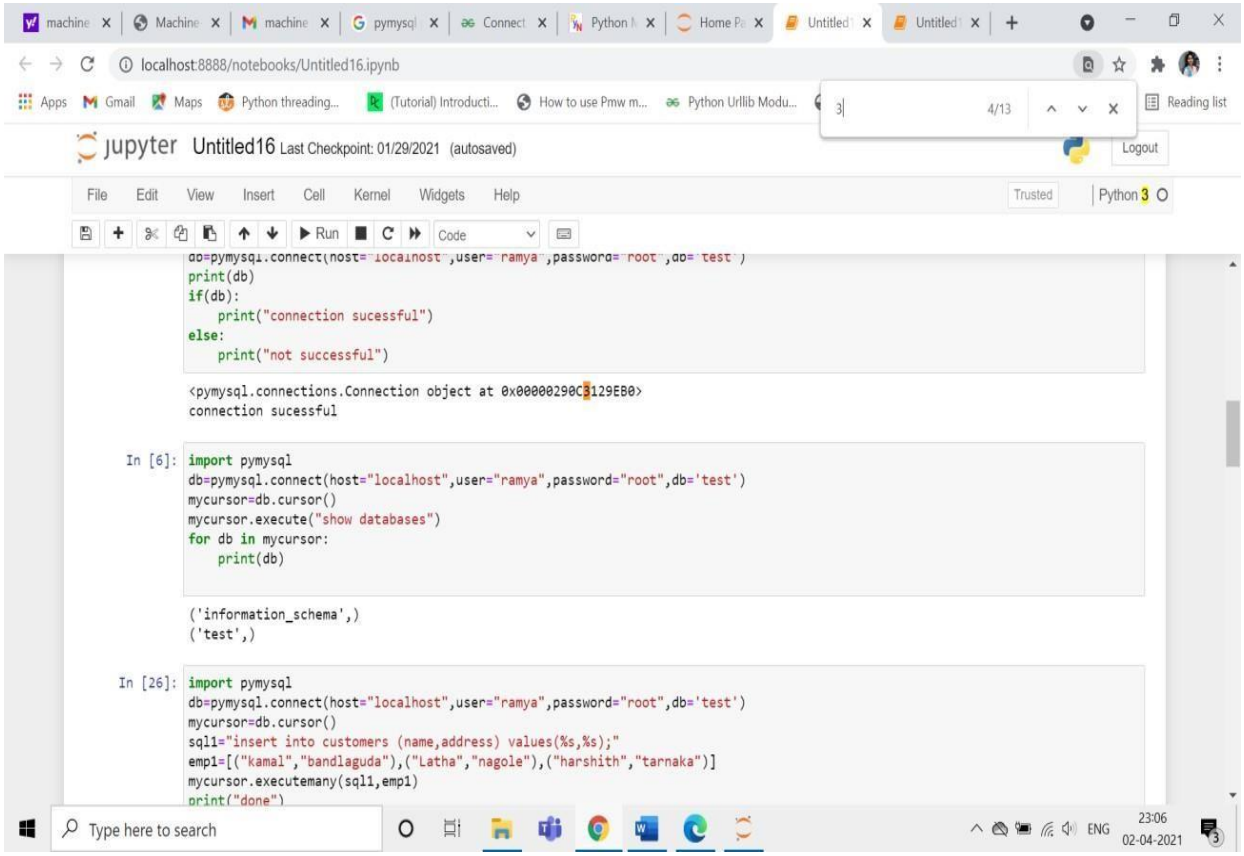
- Execute the SELECT query and process the result set returned by the queryin Python.
- Use Python **variables in a where clause** of a SELECT query to pass dynamic values.
- Use fetchall(),fetchmany(), and fetchone() methods of a cursor class to fetch all or limited rows from atable.
- Steps to fetch rows from a MySQL database table Follow these steps:–

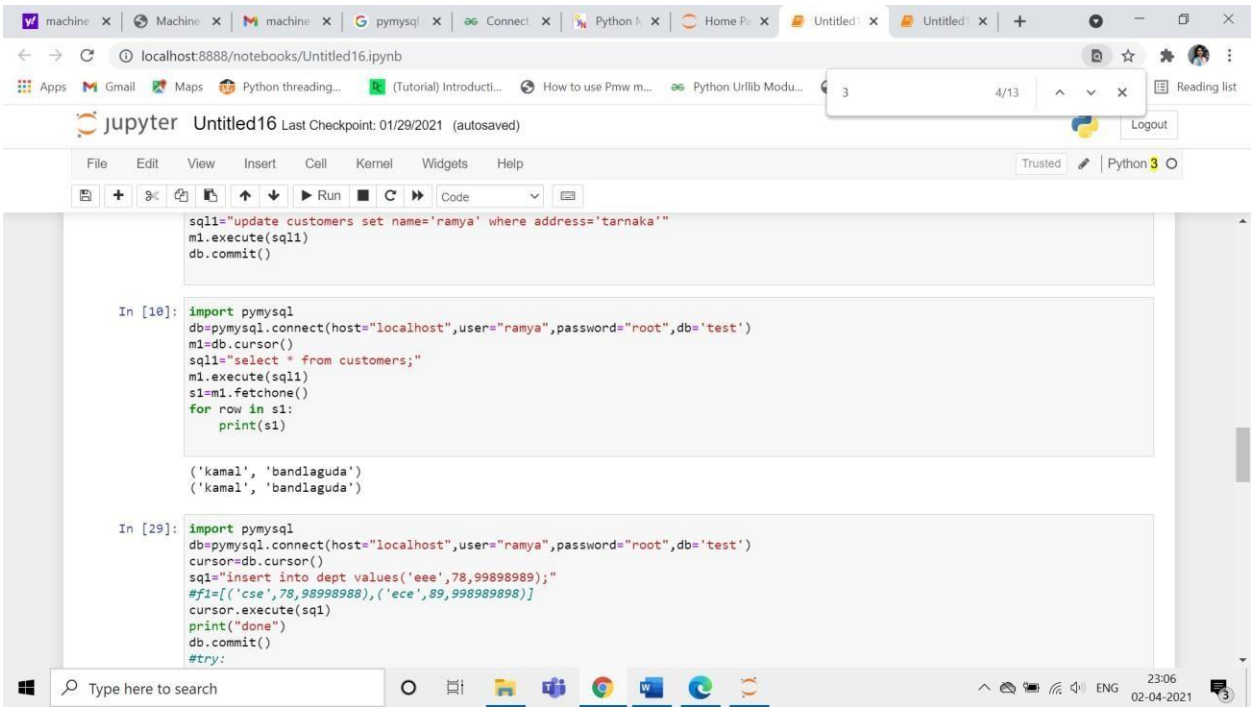
SOURCE CODE:

```
importpymysql

    defmysqlconnect():

    # To connect MySQL

database conn = pymysql.connect(

  host='localhost', user='root',
    password = "pass",
    db='College',

  )
```

```
cur = conn.cursor()
cur.execute("select @@version")
output = cur.fetchall() print(output)


# To close the connection conn.close(
)
```

OUTPUT:





VIVA QUESTIONS AND ANSWERS

1. How to select from a MySQL table using Python?

Connect to MySQL from Python

Refer to Python MySQL database connection to connect to MySQL database from Python using MySQL Connector module

Define a SQL SELECT Query

Next, prepare a SQL SELECT query to fetch rows from a table. You can select all or limited rows based on your requirement. If the where condition is used, then it decides the number of rows to fetch. For example, SELECT col1, col2,…colnN FROM MySQL_table WHERE id = 10;. This will return row number 10.

Get Cursor Object from Connection

Next, use a connection.cursor() method to create a cursor object. This method creates a new MySQLCursor object.

Execute the SELECT query using execute() method

Execute the select query using the cursor.execute() method.

Extract all rows from a result

After successfully executing a Select operation, Use the fetchall() method of a cursor object to get all rows from a query result. it returns a list of rows.

Iterate each row

Iterate a row list using a for loop and access each row individually (Access each row's column data using a column name or index number.)

Close **the cursor object and database connection object**

usecursor.clsoe() and connection.clsoe() method to close open connections after your work completes.

2. What are the methods to fetch data returned by a cursor.execute( )

- cursor.fetchall() to fetch all rows
- cursor.fetchone() to fetch a single row
- cursor.fetchmany(SIZE) to fetch limited rows

PROGRAM 3

3.IMPLEMENT K-NEAREST NEIGHBORS CLASSIFICATION USINGPYTHON

AIM : To Implement k-neighbours classification using python

DESCRIPTION:

This algorithm is used to solve the classification model problems. K-nearest neighbor or K-NN algorithm basically creates an imaginary boundary to classify the data. When new data points come in, the algorithm will try to predict that to the nearest of the boundary line.
Therefore, larger k value means smother curves of separation resulting in less complex models. Whereas, smaller k value tends to over fit the data and resulting in complex models.
It's very important to have the right k-value when analyzing the dataset to avoid over fitting and under fitting of the dataset.

KNN MODEL REPRESENTATION:

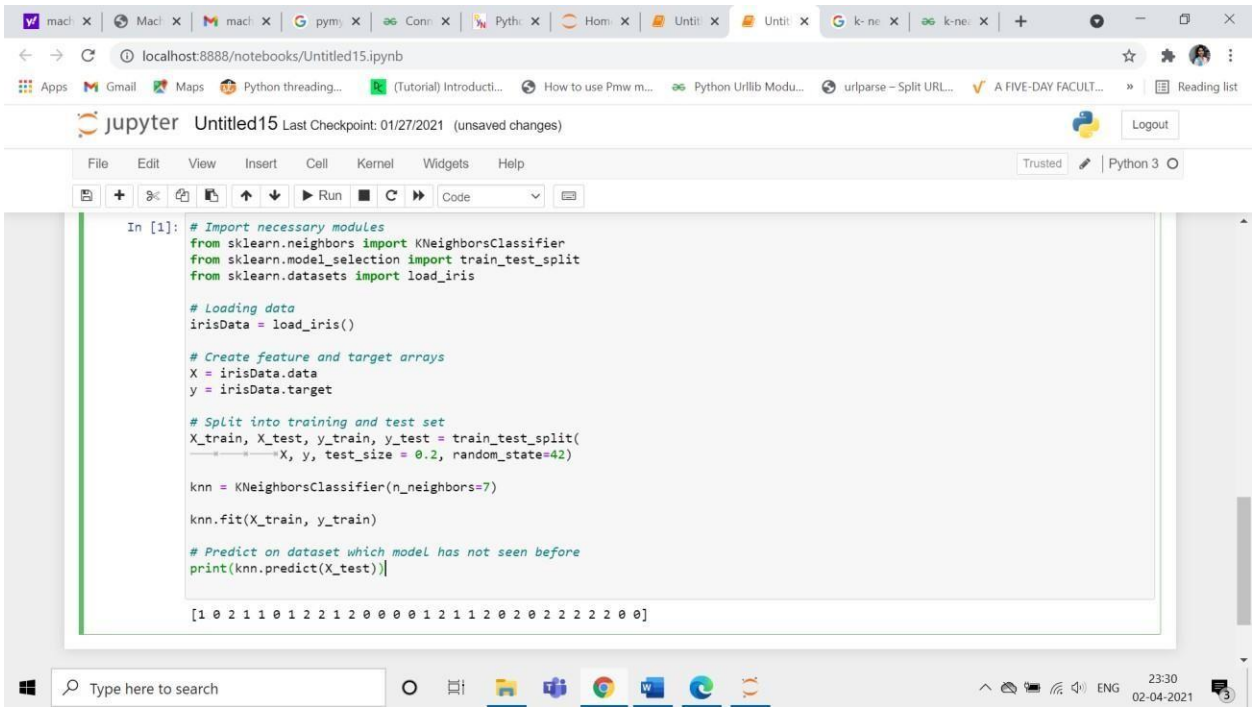
The model representation for KNN is the entire training dataset.It is as simple as that.
KNN has no model other than storing the entire dataset, so there is no learning required.
Efficient implementations can store the data using complex data structures like k-d trees to make look-up and matching of new patterns during prediction efficient.
Because the entire training dataset is stored, you may want to think carefully about the consistency of your training data. It might be a good idea to curate it, update it often as new data becomes available and remove erroneous and outlier data.

- The **k-nearest neighbor algorithm** is imported from the scikit-learnpackage.
- Create feature and target variables.
- Split data into training and testdata.
- Generate a **k-NN** model using **neighbors**value.
- Train or fit the data into themodel.
- Predict thefuture.

SOURCE CODE:

```
# Import necessarymodules
fromsklearn.neighbors importKNeighborsClassifier from
sklearn.model_selection import train_test_split from
sklearn.datasets importload_iris

# Loading data irisData =
load_iris()

# Create feature and target arrays X =
irisData.data
y = irisData.target

# Split into training and test set
X_train, X_test, y_train, y_test = train_test_split(
X, y, test_size = 0.2, random_state=42) knn =
KNeighborsClassifier(n_neighbors=7)
knn.fit(X_train, y_train)

# Predict on dataset which model has not seen before print(knn.predict(X_test))
```

3.5 OUTPUT:



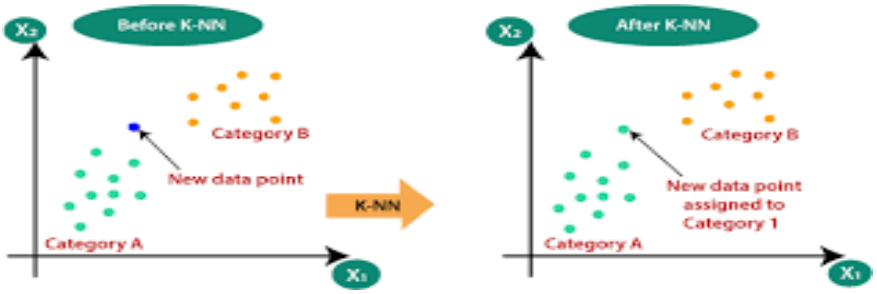
VA QUESTIONS & ANSWERS

1. What is the KNN Algorithm?

KNN(K-nearest neighbors) is a **supervised** learning and **non-parametric** algorithm that can be used to solve both classification and regression problem statements.

It uses data in which there is a target column present **i.e, labeled data** to model a function to produce an output for the unseen data. It uses the Euclidean distance formula to compute the distance between the data points for classification or prediction.

The main objective of this algorithm is that similar data points must be close to each other so it uses the distance to calculate the similar points that are close to each other.



2. Why is KNN a non-parametric Algorithm?

The term "**non-parametric"** refers to not making any assumptions on the underlying data distribution. These methods do not have any fixed numbers of parameters in the model.

Similarly in KNN, the model parameters grow with the training data by considering each training case as a parameter of the model. So, KNN is a non-parametric algorithm.

3. What is "K" in the KNN Algorithm?

K represents the number of nearest neighbors you want to select to predict the class of a given item, which is coming as an unseen dataset for the model.

4. Why is the odd value of "K" preferred over even values in the KNN Algorithm?

The odd value of K should be preferred over even values in order to ensure that there are no ties in the voting. If the square root of a number of data points is even, then add or subtract 1 to it to make it odd.

5. How does the KNN algorithm make the predictions on the unseen dataset?

The following operations have happened during each iteration of the algorithm. For each of the unseen or test data point, the kNN classifier must:

Step-1: Calculate the distances of test point to all points in the training set and store them

Step-2: Sort the calculated distances in increasing order

Step-3: Store the K nearest points from our training dataset

Step-4: Calculate the proportions of each class

Step-5: Assign the class with the highest proportion

PROGRAM-4

4. Given the following data, which specify classifications for nine combinations of VAR1 and VAR2 predict a classification for a case where VAR1=0.906 and VAR2=0.606, using the result of k-means clustering with 3 means (i.e., 3centroids)

VAR1	VAR2	CLASS
1.713	1.586	0
0.180	1.786	1
0.353	1.240	1
0.940	1.566	0
1.486	0.759	1
1.266	1.106	0
1.540	0.419	1
0.459	1.799	1
0.773	0.186	1

SOURCE CODE:

```
fromsklearn.cluster import KMeans

import numpy as np

X = np.array([[1.713,1.586], [0.180,1.786], [0.353,1.240],

[0.940,1.566], [1.486,0.759], [1.266,1.106],[1.540,0.419],[0.459,1.799],[0.773,0.186]])

y=np.array([0,1,1,0,1,0,1,1,1])

kmeans = KMeans(n_clusters=3, random_state=0).fit(X,y)
```

4.2.OUTPUT:

```
kmeans.predict([[0.906, 0.606]])
```

4.3 VIVA QUESTIONS & ANSWERS

1. What is K means Clustering Algorithm?

K Means algorithm is a centroid-based clustering (unsupervised) technique. This technique groups the dataset into k different clusters having an almost equal number of points. Each of the clusters has a centroid point which represents the mean of the data points lying in that cluster.

The idea of the K-Means algorithm is to find k-centroid points and every point in the dataset will belong to either of the k-sets having minimum Euclidean distance.

2. Is Feature Scaling required for the K means Algorithm?

Yes, K-Means typically needs to have some form of normalization done on the datasets to work properly since it is sensitive to both the mean and variance of the datasets.

For performing feature scaling, generally, **StandardScaler** is recommended, but depending on the specific use cases, other techniques might be more suitable as well.

For Example, let's have 2 variables, named age and salary where age is in the range of 20 to 60 and salary is in the range of 100-150K, since scales of these variables are different so when these variables are substituted in the euclidean distance formula, then the variable which is on the large scale suppresses the variable which is on the smaller scale. So, the impact of age will not be captured very clearly. Hence, you have to scale the variables to the same range using **Standard Scaler, Min-Max Scaler**, etc.

3. Which metrics can you use to find the accuracy of the K means Algorithm?

There does not exist a correct answer to this question as k means being an unsupervised learning technique does not discuss anything about the output column. As a result, one can not get the accuracy number or values from the algorithm directly.

4. What are the advantages and disadvantages of the K means Algorithm?

Advantages:

- Easy to understand and implement.
- Computationally efficient for both training and prediction.
- Guaranteed convergence.

Disadvantages:

- We need to provide the number of clusters as an input variable to the algorithm.
- It is very sensitive to the initialization process.
- Good at clustering when we are dealing with spherical cluster shapes, but it will perform poorly when dealing with more complicated shapes.
- Due to the leveraging of the Euclidean distance function, it is sensitive to outliers.

5. What are the ways to avoid the problem of initialization sensitivity in the K means Algorithm?

There are two ways to avoid the problem of initialization sensitivity:

- **Repeat K means:** It basically repeats the algorithm again and again along with initializing the centroids followed by picking up the cluster which results in the small intracluster distance and large

intercluster distance.

- **K Means++:** It is a smart centroid initialization technique.

Amongst the above two techniques, K-Means++ is the best approach.

PROGRAM 5

5.The Following Training Examples Map Descriptions Of Individuals Onto High, Medium And LowCredit-Worthiness.

medium skiing design single twenties no ->highRisk

high golf trading married forties yes ->lowRisk

low speedway transport married thirties yes ->medRisk

medium football banking single thirties yes ->lowRisk

high flying media married fifties yes ->highRisk

low football security single twenties no ->medRisk

medium golf media single thirties yes ->medRisk

medium golf transport married forties yes ->lowRisk

high skiing banking single thirties yes ->highRisk low

golf unemployed married forties yes ->highRisk

SOURCE CODE:

Input attributes are (from left to right) income, recreation, job, status, age-group, home-owner.Findtheunconditionalprobabilityof`golf'andtheconditionalprobabilityofsingle' given `medRisk' in thedataset?

totalRecords=10

numberGolfRecreation=4

probGolf=numberGolfRecreation/totalRecordsprint("Unconditio

nal probability of golf: ={}".format(probGolf)) #conditional

probability of `single' given`medRisk'

bayes Formula

#p(single|medRisk)=p(medRisk|single)p(single)/p(medRisk)

#p(medRisk|single)=p(medRisk ∩single)/p(single)

```
# Therefore the result is:

numberMedRiskSingle=2

numberMedRisk=3

probMedRiskSingle=numberMedRiskSingle/totalRecordspr

obMedRisk=numberMedRisk/totalRecordsconditionalProba

bility=(probMedRiskSingle/probMedRisk)

print("Conditional probability of single given medRisk: = {}".format(conditionalProbability))
```

OUTPUT:

```
Unconditional probability of golf: = 0.4

Conditional probability of single given medRisk: = 0.6666666666666667
```

VIVA QUESTIONS & ANSWERS:

1. How K means++ clustering Algorithm works?

K Means++ algorithm is a smart technique for centroid initialization that initialized one centroid while ensuring the others to be far away from the chosen one resulting in faster convergence.

The steps to follow for centroid initialization are:

Step-1: Pick the first centroid point randomly.

Step-2: Compute the distance of all points in the dataset from the selected centroid. The distance of x_i point from the farthest centroid can be calculated by the given formula:

$$d_i = max_{(j:1 \mapsto m)} ||x_i - C_j||^2$$

where,

d_i: Distance of x_i point from the farthest centroid

m: number of centroids already picked

Step-3: Make the point x_i as the new centroid that is having maximum probability proportional to d_i

Step-4: Repeat the above last two steps till you find k centroids.

2. What is the training and testing complexity of the K means Algorithm?

Training complexity in terms of Big-O notation:

If we use Lloyd's algorithm, the complexity for training is: **"K*I*N*M"**

where,

K: It represents the number of clusters

I: It represents the number of iterations

N: It represents the sample size

M: It represents the number of variables

Conclusion: There is a significant Impact on capping the number of iterations.

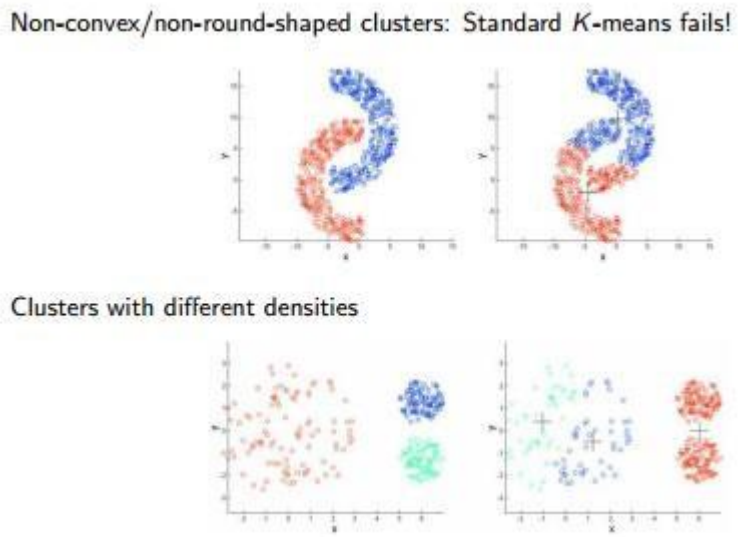
Predicting complexity in terms of Big-O notation:

"K*N*M"

Prediction needs to be computed for each record, the distance to each cluster and assigned to the nearest ones.

3. Is it possible that the assignment of data points to clusters does not change between successive iterations in the K means Algorithm?

When the K-Means algorithm has reached the local or global minima, it will not change the assignment of data points to clusters for two successive iterations during the algorithm run.



4. How to Choose K Value in K-Means:

1. Elbow method

steps:

step1: compute clustering algorithm for different values of k.

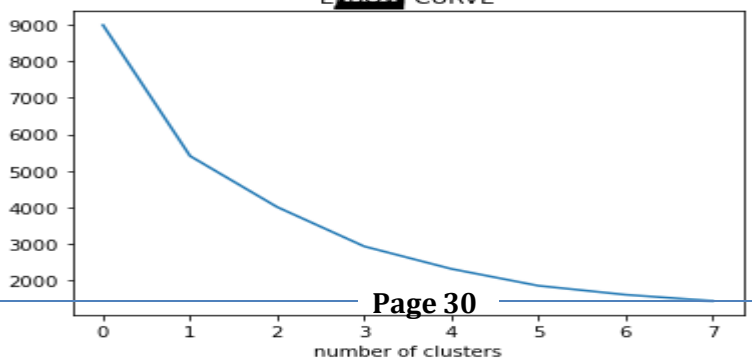
for example k=[1,2,3,4,5,6,7,8,9,10]

step2: for each k calculate the within-cluster sum of squares(WCSS).

step3: plot curve of WCSS according to the number of clusters.

step4: The location of bend in the plot is generally considered an indicator of the approximate number of clusters.

$$WCSS = \sum (X_i - Y_i)^2$$



5. What are the practical Considerations In K-Means:

- A choosing number of Clusters in Advance (K).
- Standardization of Data (scaling).
- Categorical Data (can be solved with K-Mode).
- Impact of initial Centroids and Outliers.

6. Why do you prefer Euclidean distance over Manhattan distance in the K means Algorithm?

Euclidean distance is preferred over Manhattan distance since Manhattan distance calculates distance only vertically or horizontally due to which it has dimension restrictions.

On the contrary, Euclidean distance can be used in any space to calculate the distances between the data points. Since in K means algorithm the data points can be present in any dimension, so Euclidean distance is a more suitable option.

PROGRAM 6:

6. IMPLEMENT LINEAR REGRESSION USINGPYTHON.

AIM:

To Implement linear regression using python.

DESCRIPTION:

Regression: Regression analysis is one of the most important fields in statistics and machine learning.There are many regression methods available. Linear regression is one of them

What Is Regression?

Regression analysis is one of the most important fields in statistics and machine learning.Thereare many regression methods available. Linear regression is one of them.Regression searches for relationships among variables.For example, you can observe several employees of some company and try to understand how their salaries depend on the **features**, such as experience, level of education, role, city they work in, and so on.This is a regression problem where data related to each employee represent one **observation**.The presumption is that the experience, education, role, and city are the independent features, while the salary depends onthem.Generally, in regression analysis, you usually consider some phenomenon of interest andTherehave a number of observations. Each observation has two or more features. Following the assumption that(atleast)oneofthefeaturesdependsontheothers,youtrytoestablisharelationamongthem.you need to find a function that maps some features or variables to others sufficiently well. The dependent features are called the **dependent variables**, **outputs**, or **responses**.The independent features are called the **independent variables**, **inputs**, or **predictors**.

LinearRegression:

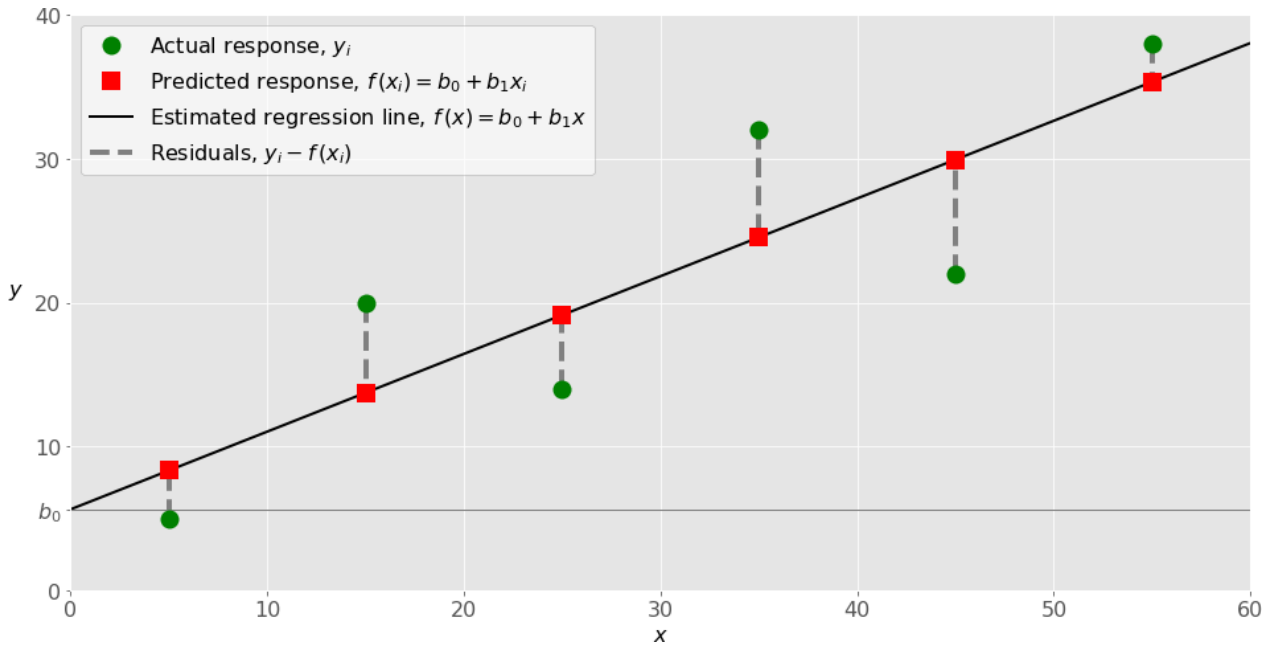
Linear regression is probably one of the most important and widely used regression techniques. It's among the simplest regression methods. One of its main advantages is the ease of interpretingresults.When implementing linear regression of some dependent variable *y*on the set of independent variables x= ($x_{1...}x_r$), where ris the number of predictors, you assume a linear relationship between y andx: $y= \beta_0 + \beta_1x_1 + \ldots\ldots+ \beta_rx_r+ \epsilon$. This equation is the **regression equation**. β_0,β, $\beta_r$arethe**regressioncoefficients**, and $\varepsilon$is the**randomerror**.

Linear regression calculates the **estimators** of the regression coefficients or simply the **predicted weights**, denoted with $b_1 \ldots b_r$. They define the **estimated regressionfunction f**(x) = b0 +b_1x_1 +

….+$b_r x_r$ This function should capture the dependencies between the inputsand output sufficiently well.

Simple Linear Regression

The following figure illustrates simple linear regression:



When implementing simple linear regression, you typically start with a given set of input-output (x-y) pairs (green circles). These pairs are your observations. For example, the leftmost observation (green circle) has the input x= 5 and the actual output (response) y= 5. The next one has x= 15 and y= 20, and so on.

Theestimatedregressionfunction(blackline)hastheequationf(x)=b_0+b1x.Yourgoalistocalculatetheoptimalvaluesofthepredictedweights$b_0$andb1thatminimizeSSRanddetermine theestimatedregressionfunction.Thevalueofb_0,alsocalledthe**intercept**,showsthepointwheretheestimatedregressionlinecrossesthe y axis.Itisthevalueoftheestimatedresponse f(x)for x=0. Thevalue of b1 determinesthe**slope**of theestimated regression line.

The predicted responses (red squares) are the points on the regression line that correspond to the input values. For example, for the input x= 5, the predicted response is f (5) = 8.33 (represented with the leftmost red square).

Theresiduals(verticaldashedgraylines)canbecalculatedasy_1-f(x_1)=y_1–b_0–$b_1x_1$fori=1,

…,n . They are the distances between the green circles and red squares. When you implement linear regression, you are actually trying to minimize these distances and make the red squares as close to the predefined green circles as possible.

Implementing Linear Regression in Python

It's time to start implementing linear regression in Python. Basically, all you should do is apply the proper packages and their functions and classes.

Python Packages for Linear Regression

The package **NumPy**is a fundamental Python scientific package that allows many high-performanceoperationsonsingle-andmulti-dimensionalarrays.Italsooffersmanymathematical routines. Of course, it's opensource.

The package **scikit-learn** is a widely used Python library for machine learning, built on top of NumPy and some other packages. It provides the means for preprocessing data, reducing dimensionality,implementingregression,classification,clustering,andmore.LikeNumPy,scikit-learn is also opensource.

If you want to implement linear regression and need the functionality beyond the scope of scikit-learn, you should consider **statsmodels**. It's a powerful Python package for the estimation of statistical models, performing tests, and more. It's open source as well.

Simple Linear Regression With scikit-learn

Let's start with the simplest case, which is simple linear regression. There

are five basic steps when you're implementing linear regression:

1. Import the packages and classes youneed.
2. Provide data to work with and eventually do appropriatetransformations.
3. Create a regression model and fit it with existingdata.
4. Check the results of model fitting to know whether the model issatisfactory.
5. Apply the model forpredictions.

SOURCE CODE:

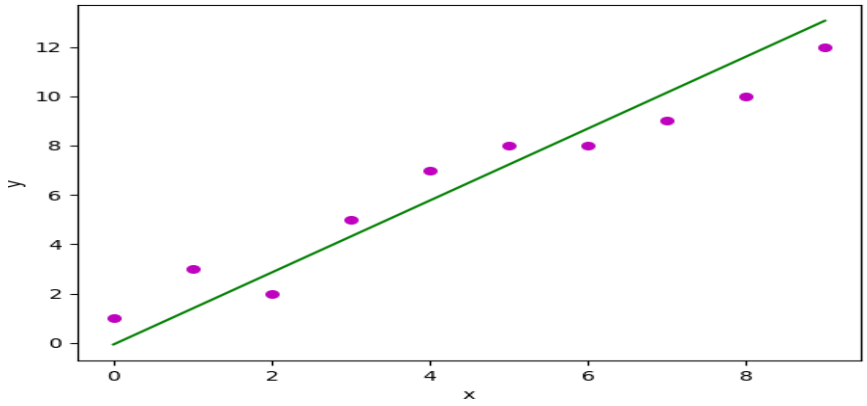
```
importnumpy as np

importmatplotlib.pyplot as plt


defestimate_coef(x, y):

# number of observations/points n =
```

```
np.size(x)

# mean of x and y vector
m_x, m_y = np.mean(x), np.mean(y)

# calculating cross-deviation and deviation aboutx SS_xy =

np.sum(y*x) - n*m_y*m_x

SS_xx = np.sum(x*x) -n*m_x*m_x

# calculating regression coefficients b_1 =

SS_xy / SS_xx

b_0 = m_y - b_1*m_x

return(b_0, b_1)

defplot_regression_line(x, y, b):

# plotting the actual points as scatter plot

plt.scatter(x, y, color = "m",

marker = "o", s = 30)

# predicted response vector y_pred =

b[0] + b[1]*x

# plotting the regression line

plt.plot(x, y_pred, color = "g")

# putting labels
plt.xlabel('x')

plt.ylabel('y')
```

```
# function to show plot

plt.show()

def main():

# observations

x = np.array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])

y = np.array([1, 3, 2, 5, 7, 8, 8, 9, 10, 12])


# estimating coefficients b =

estimate_coef(x, y)

print("Estimated coefficients:\nb_0 = {} \

\nb_1 = {}".format(b[0], b[1]))


# plotting regression line

plot_regression_line(x, y, b)


ifname_____== "main":

main()
```

OUTPUT:

```
Estimated coefficients:
b_0 = -0.05862068965
```



VIVA QUESTIONS AND ANSWERS

1. **What do you mean by the Logistic Regression?**

It's a classification algorithm that is used where the target variable is of categorical nature. The main objective behind Logistic Regression is to determine the relationship between features and the probability of a particular outcome.

For Example, when we need to predict whether a student passes or fails in an exam given the number of hours spent studying as a feature, the target variable comprises two values i.e. pass and fail.

Therefore, we can solve classification problem statements which is a supervised machine learning technique using Logistic Regression.

2. **What are the different types of Logistic Regression?**

Three different types of Logistic Regression are as follows:

1. Binary Logistic Regression: In this, the target variable has only two 2 possible outcomes.

For Example, 0 and 1, or pass and fail or true and false.

2. Multinomial Logistic Regression: In this, the target variable can have three or more possible values without any order.

For Example, Predicting preference of food i.e. Veg, Non-Veg, Vegan.

3. Ordinal Logistic Regression: In this, the target variable can have three or more values with ordering.

For Example, Movie rating from 1 to 5.

***3.* How do we handle categorical variables in Logistic Regression?**

The inputs given to a Logistic Regression model need to be numeric. The algorithm cannot handle categorical variables directly. So, we need to convert the categorical data into a numerical format that is suitable for the algorithm to process.

Each level of the categorical variable will be assigned a unique numeric value also known as a **dummy variable.** These dummy variables are handled by the Logistic Regression model in the same manner as any other numeric value.

***4.* What are the assumptions made in Logistic Regression?**

Some of the assumptions of Logistic Regression are as follows:

1. It assumes that there is minimal or **no multi collinearity** among the independent variables i.e, predictors are not correlated.

2. There should be a linear relationship between the logit of the outcome and each predictor variable. The logit function is described as **logit(p) = log(p/(1-p))**, where p is the probability of the target outcome.

3. Sometimes to predict properly, it usually requires a **large sample size**.

4. The Logistic Regression which has **binary classification**i.e, two classes assume that the target variable is binary, and ordered Logistic Regression requires the target variable to be ordered.

For example, Too Little, About Right, Too Much.

5. It assumes there is **no dependency** between the observations.

***5.* Can we solve the multiclass classification problems using Logistic Regression? If Yes then How?**

Yes, in order to deal with multiclass classification using Logistic Regression, the most famous method is known as the one-vs-all approach. In this approach, a number of models are trained, which is equal to the number of classes. These models work in a specific way.

For Example, the first model classifies the data point depending on whether it belongs to class 1 or some other class(not class 1); the second model classifies the data point into class 2 or some other class(not class 2) and so-on for all other classes. So, in this manner, each data point can be checked over all the classes.

PROGRAM 7

IMPLEMENT NAÏVE BAYES THEOREM TO CLASSIFY THE ENGLISHTEXT

AIM:

To Implement naïve baye's theorem to classify the English text

DESCRIPTION:

The challenge of text classification is to attach labels to bodies of text, e.g., tax document, medical form, etc. based on the text itself. For example, think of your spam folder in your email. How does your email provider know that a particular message is spam or "ham" (not spam)? We'll take a look at one natural language processing technique for text classification called Naive Bayes

SOURCE CODE:

```
import pandas as pd

fromsklearn.model_selection import train_test_split

fromsklearn.feature_extraction.text import CountVectorizer

from sklearn.naive_bayes import MultinomialNB

fromsklearn.metrics import accuracy_score, confusion_matrix, precision_score, recall_score


msg = pd.read_csv('document.csv', names=['message', 'label'])

print("Total Instances of Dataset: ", msg.shape[0])

msg['labelnum'] = msg.label.map({'pos': 1, 'neg': 0})


X = msg.message

y = msg.labelnum


Xtrain, Xtest, ytrain, ytest = train_test_split(X, y)


count_v = CountVectorizer()
```

```
Xtrain_dm = count_v.fit_transform(Xtrain)

Xtest_dm = count_v.transform(Xtest)

df = pd.DataFrame(Xtrain_dm.toarray(),columns=count_v.get_feature_names())

clf = MultinomialNB()

clf.fit(Xtrain_dm, ytrain)

pred = clf.predict(Xtest_dm)

print('Accuracy Metrics:')

print('Accuracy: ', accuracy_score(ytest, pred)) print('Recall: ', recall_score(ytest, pred)) print('Precision: ',

precision_score(ytest, pred))

print('Confusion Matrix: \n', confusion_matrix(ytest, pred))
```

document.csv:

```
I love this sandwich,pos This

is an amazingplace,pos

I feel very good about these beers,pos

This is my best work,pos

What an awesome view,pos

I do not like this restaurant,neg

I am tired of this stuff,neg

I can't deal with this,neg He

is my sworn enemy,neg My
```

boss is horrible,neg

This is an awesome place,pos

I do not like the taste of this juice,neg

I love to dance,pos

I am sick and tired of this place,neg

What a great holiday,pos

That is a bad locality to stay,neg
We will have good fun tomorrow,pos I

went to my enemy's house today,neg

OUTPUT:

Total Instances of Dataset: 18

Accuracy Metrics:

Accuracy: 0.6

Recall: 0.6666666666666666

Precision: 0.6666666666666666

Confusion Matrix:

[[1 1]

[1 2]]

VIVA QUESTIONS & ANSWERS

1. How Naive Bayes algorithm works?

Let's understand it using an example. Below I have a training data set of weather and corresponding target variable 'Play' (suggesting possibilities of playing). Now, we need to classify whether players will play or not based on weather condition. Let's follow the below steps to perform it.

Step 1: Convert the data set into a frequency table

Step 2: Create Likelihood table by finding the probabilities like Overcast probability = 0.29 and probability of playing is 0.64.

Weather	Play
Sunny	No
Overcast	Yes
Rainy	Yes
Sunny	Yes
Sunny	Yes
Overcast	Yes
Rainy	No
Rainy	No
Sunny	Yes
Rainy	Yes
Sunny	No
Overcast	Yes
Overcast	Yes
Rainy	No

Frequency Table		
Weather	No	Yes
Overcast		4
Rainy	3	2
Sunny	2	3
Grand Total	5	9

Likelihood table				
Weather	No	Yes		
Overcast		4	=4/14	0.29
Rainy	3	2	=5/14	0.36
Sunny	2	3	=5/14	0.36
All	5	9		
	=5/14	=9/14		
	0.36	0.64		

Step 3: Now, use NaiveBayesian equation to calculate the posterior probability for each class. The class with the highest posterior probability is the outcome of prediction.

Problem: Players will play if weather is sunny. Is this statement is correct?

We can solve it using above discussed method of posterior probability.

P(Yes | Sunny) = P(Sunny | Yes) * P(Yes) / P (Sunny)

Here we have P (Sunny |Yes) = 3/9 = 0.33, P(Sunny) = 5/14 = 0.36, P(Yes)= 9/14 = 0.64

Now, P (Yes | Sunny) = 0.33 * 0.64 / 0.36 = 0.60, which has higher probability.

Naive Bayes uses a similar method to predict the probability of different class based on various attributes. This algorithm is mostly used in text classification and with problems having multiple classes.

2. Applications of Naive Bayes Algorithms

- **Real time Prediction:** Naive Bayes is an eager learning classifier and it is sure fast. Thus, it could be used for making predictions in real time.
- **Multi class Prediction:** This algorithm is also well known for multi class prediction feature. Here we can predict the probability of multiple classes of target variable.
- **Text classification/ Spam Filtering/ Sentiment Analysis:** Naive Bayes classifiers mostly used in text classification (due to better result in multi class problems and independence rule) have higher success rate as compared to other algorithms. As a result, it is widely used in Spam filtering (identify spam e-mail) and Sentiment Analysis (in social media analysis, to identify positive and negative customer sentiments)
- **Recommendation System:** Naive Bayes Classifier and Collaborative Filtering together builds a Recommendation System that uses machine learning and data mining techniques to filter unseen information and predict whether a user would like a given resource or not

PROGRAM 8

8.IMPLEMENT AN ALGORITHM TO DEMONSTRATE THE SIGNIFICANCE OF GENETICALGORITHM

AIM:

To Implement an algorithm to demonstrate the significance of genetic algorithm

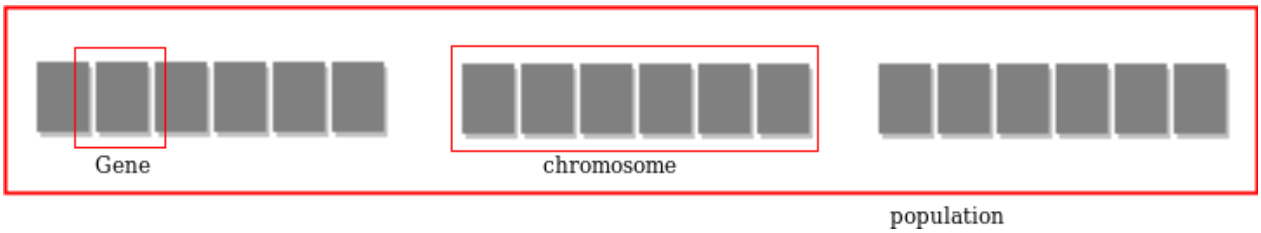
DESCRIPTION:

Genetic Algorithms (GAs) are adaptive heuristic search algorithms that belong to the larger part of evolutionary algorithms. Genetic algorithms are based on the ideas of natural selection and genetics. These are intelligent exploitation of random search provided with historical data to direct the search into the region of better performance in solution space. They are commonly used to generate high-quality solutions for optimization problems and search problems.Genetic algorithms simulate the process of natural selection which means those species who can adapt to changes in their environment are able to survive and reproduce and go to next generation. In simple words, they simulate "survival of the fittest" among individual of consecutive generation for solving a problem. Each generation consist of a population of individuals and each individual represents a point in search space and possible solution. Each individual is represented as a string of character/integer/float/bits. This string is analogous to the Chromosome.Genetic algorithms are based on an analogy with genetic structure and behavior of chromosome of the population. Following is the foundation of GAs based on this analogy –

- Individual in population compete for resources and mate
- Those individuals who are successful (fittest) then mate to create more offspring than others
- Genes from "fittest" parent propagate throughout the generation, that is sometimes parents create offspring which is better than either parent.
- Thus each successive generation is more suited for their environment.

The population of individuals are maintained within search space. Each individual represent a solution in search space for given problem. Each individual is coded as a finite length vector (analogous to chromosome) of components. These variable components are

analogous to Genes. Thus a chromosome (individual) is composed of several genes (variable components)



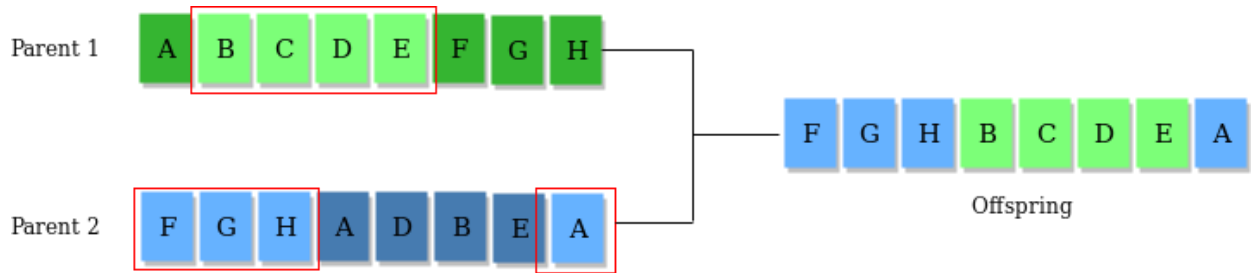
A Fitness Score is given to each individual which **shows the ability of an individual to "compete"**. The individual having optimal fitness score (or near optimal) are sought.The GAs maintains the population of n individuals (chromosome/solutions) along with their fitness scores.The individuals having better fitness scores are given more chance to reproduce than others. The individuals with better fitness scores are selected who mate and produce **better offspring** by combining chromosomes of parents. The population size is static so the room has to be created for new arrivals. So, some individuals die and get replaced by new arrivals eventually creating new generation when all the mating opportunity of the old population is exhausted. It is hoped that over successive generations better solutions will arrive while least fit die.Each new generation has on average more "better genes" than the individual (solution) of previous generations. Thus each new generations have better **"partial solutions"** than previous generations. Once the offsprings produced having no significant difference than offspring produced by previous populations, the population is converged. The algorithm is said to be converged to a set of solutions for the problem.

OPERATORS OF GENETIC ALGORITHM:

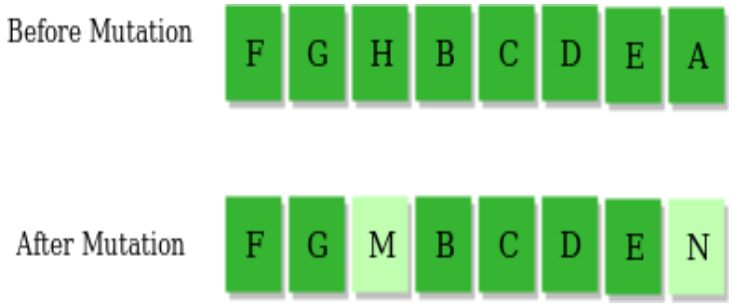
Once the initial generation is created, the algorithm evolve the generation using following operators –

1) Selection Operator: The idea is to give preference to the individuals with good fitness scores and allow them to pass there genes to the successive generations.

2) Crossover Operator: This represents mating between individuals. Two individuals are selected using selection operator and crossover sites are chosen randomly. Then the genes at these crossover sites are exchanged thus creating a completely new individual (offspring). For example –



Mutation Operator: The key idea is to insert random genes in offspring to maintain the diversity in population to avoid the premature convergence. For example –



ALGORITHM:

1) Randomly initialize populations p
2) Determine fitness of population
3) Untill convergence repeat:
   a) Select parents from population
   b) Crossover and generate new population
   c) Perform mutation on new population
   d) Calculate fitness for new population

USES OF GENETIC ALGORITHM:

- They are Robust
- Provide optimisation over large space state.
- Unlike traditional AI, they do not break on slight change in input or presence of noise

SOURCE CODE:

```
importnumpy

defcal_pop_fitness(equation_inputs, pop):

# Calculating the fitness value of each solution in the current population.

# The fitness function calulates the sum of products between each input and its
corresponding weight.

fitness = numpy.sum(pop*equation_inputs,

axis=1) return fitness


defselect_mating_pool(pop, fitness, num_parents):

# Selecting the best individuals in the current generation as parents for producing the
offspring of the next generation.

parents = numpy.empty((num_parents,

pop.shape[1])) for parent_num in

range(num_parents):

max_fitness_idx = numpy.where(fitness ==

numpy.max(fitness)) max_fitness_idx =

max_fitness_idx[0][0]

parents[parent_num, :] =

pop[max_fitness_idx, :]

fitness[max_fitness_idx] = -99999999999

return parents
def crossover(parents,

offspring_size): offspring =

numpy.empty(offspring_size)
```

```
# The point at which crossover takes place between two parents. Usually, it is at the center.
crossover_point = numpy.uint8(offspring_size[1]/2)
for k in range(offspring_size[0]):
# Index of the first parent to
mate. parent1_idx =
k%parents.shape[0]
# Index of the second parent to mate.
parent2_idx = (k+1)%parents.shape[0]
# The new offspring will have its first half of its genes taken from the first
parent. offspring[k, 0:crossover_point] = parents[parent1_idx,
0:crossover_point]
# The new offspring will have its second half of its genes taken from the second parent.
offspring[k, crossover_point:] = parents[parent2_idx, crossover_point:]
return offspring

def mutation(offspring_crossover, num_mutations=1):
mutations_counter = numpy.uint8(offspring_crossover.shape[1] / num_mutations)
# Mutation changes a number of genes as defined by the num_mutations
argument. The changes are random.
foridx in
range(offspring_crossover.shape[0]):
gene_idx = mutations_counter - 1
formutation_num in range(num_mutations):
# The random value to be added to the gene.
```

```
random_value = numpy.random.uniform(-1.0, 1.0,

1)

offspring_crossover[idx, gene_idx] = offspring_crossover[idx,
gene_idx] + random_value

gene_idx = gene_idx +

mutations_counter return

offspring_crossover

importnumpy

"""

The y=target is to maximize this

equation ASAP: y =

w1x1+w2x2+w3x3+w4x4+w5x5+6wx6

where (x1,x2,x3,x4,x5,x6)=(4,-2,3.5,5,-

11,-4.7)

What are the best values for the 6 weights w1 to w6?

We are going to use the genetic algorithm for the best possible values after a
number of generations.

"""

# Inputs of the equation.

equation_inputs = [4,-

2,3.5,5,-11,-4.7]

# Number of the weights we are looking to

optimize. num_weights = len(equation_inputs)
```

"""

Genetic algorithm

parameters: Mating

pool size

Population size

"""

sol_per_pop = 8

num_parents_mating = 4
# Defining the population size.

pop_size = (sol_per_pop,num_weights) # The population will have sol_per_pop chromosome where each chromosome has num_weights genes.

#Creating the initial population.

new_population = numpy.random.uniform(low=-4.0, high=4.0,

size=pop_size) print(new_population)

"""

new_population[0, :] = [2.4, 0.7, 8, -2, 5, 1.1]

new_population[1, :] = [-0.4, 2.7, 5, -1, 7, 0.1]

new_population[2, :] = [-1, 2, 2, -3, 2,0.9]

new_population[3, :]=[4, 7, 12, 6.1, 1.4, -4]

new_population[4, :] = [3.1, 4, 0, 2.4, 4.8,0]

new_population[5, :] = [-2, 3, -

7,6, 3, 3] """

```
best_outputs = []
num_generations =
1000
for generation in
range(num_generations):
print("Generation : ",
generation)
# Measuring the fitness of each chromosome in the
population. fitness = cal_pop_fitness(equation_inputs,
new_population) print("Fitness")
print(fitness)
best_outputs.append(numpy.max(numpy.sum(new_population*equation_inputs,
axis=1))) # The best result in the current iteration.
print("Best result : ", numpy.max(numpy.sum(new_population*equation_inputs, axis=1)))

# Selecting the best parents in the population for
mating. parents =
select_mating_pool(new_population, fitness,
num_parents_mating)
print("Parents
")
print(parents)

# Generating next generation using
```

```
crossover. offspring_crossover =
crossover(parents,
offspring_size=(pop_size[0]-parents.shape[0], num_weights))
print("Crossover")
print(offspring_cr
ossover)

# Adding some variations to the offspring using mutation.
offspring_mutation = mutation(offspring_crossover,
num_mutations=2) print("Mutation")
print(offspring_mutation)

# Creating the new population based on the parents and offspring.
new_population[0:parents.shape[0], :] = parents
new_population[parents.shape[0]:, :] = offspring_mutation
# Getting the best solution after iterating finishing all generations.
#At first, the fitness is calculated for each solution in the final
generation. fitness = cal_pop_fitness(equation_inputs,
new_population)
# Then return the index of that solution corresponding to the
best fitness. best_match_idx = numpy.where(fitness ==
numpy.max(fitness))

print("Best solution : ",
```

```
new_population[best_match_idx, :]) print("Best
solution fitness : ", fitness[best_match_idx])
importmatplotlib.pyplotmat
plotlib.pyplot.plot(best_outp
uts)
matplotlib.pyplot.xlabel("Ite
ration")
matplotlib.pyplot.ylabel("Fit
ness")
matplotlib.pyplot.show()
```

OUTPUT:

```
[[ 0.58204141 2.32880696 -2.95130209 2.57056953 3.33055238 -0.58167871] [-1.65052225
3.52263842 -2.46577305 -1.7005396 -3.80480202 0.29677167] [ 2.6239874 -2.01548549 -
1.72292295 3.61090243 -1.25604726 -2.32647264] [-3.45167393 2.85771825 3.74655682 -
2.01790626 0.25750106 -3.12923247] [ 2.86026334 -0.4306777 -3.26297956 1.74863348-
1.93705571 -3.18855672] [-1.70012089 0.98685104 -1.91192072 3.91873942-0.09354385
1.43038667] [ 0.31769009 -0.87290809 3.75249785 2.57657993 0.58883082 2.83231871] [
3.83314926 0.33838112 -2.49509594 -1.50763174 3.99440509 -0.03037715]]
Gen
erati
on :
0
Fitn
ess
[-33.70834413 9.67772594 51.30214363 -4.62383365 45.91897711 -1.56604606 9.24418172 -
```

45.41084308]

Best result :

51.302143629097614

Parents

[[2.6239874 -2.01548549 -1.72292295 3.61090243 -1.25604726 -2.32647264] [2.86026334 -0.4306777 -3.26297956 1.74863348 -1.93705571 -3.18855672] [-1.65052225 3.52263842 -2.46577305 -1.7005396 -3.80480202 0.29677167] [0.31769009 -0.87290809 3.75249785 2.57657993 0.58883082 2.83231871]]

Crossover

[[2.6239874 -2.01548549 -1.72292295 1.74863348 -1.93705571 -3.18855672] [2.86026334 -0.4306777 -3.26297956 -1.7005396 -3.80480202 0.29677167] [-1.65052225 3.52263842 -2.46577305 2.57657993 0.58883082 2.83231871] [0.31769009 -0.87290809 3.75249785 3.61090243 -1.25604726 -2.32647264]]

Mutation

[[2.6239874 -2.01548549 -1.67896632 1.74863348 -1.93705571 -3.97789372] [2.86026334 -0.4306777 -3.12878279 -1.7005396 -3.80480202 -0.15430324] [-1.65052225 3.52263842 -3.37669601 2.57657993 0.58883082 2.25153466] [0.31769009 -0.87290809 2.93428907 3.61090243 -1.25604726 -2.71597954]]

.

.

.

.

Gener

ation :

999

Fitnes

s

[2554.3935562 2551.72360738 2549.40583954 2549.29931629 2552.24225166
2550.45506206
2547.1299512 2551.22467397]

Best result :

2554.3935561987346

Parents

[[3.17690088e-01 -8.72908094e-01 2.67689952e+02 1.74863348e+00 -
1.93705571e+00 - 3.37108802e+02] [3.17690088e-01 -8.72908094e-01
2.67638232e+02 1.74863348e+00 -
1.93705571e+00 -3.36689592e+02] [3.17690088e-01 -8.72908094e-01 2.67254110e+02
1.74863348e+00 -1.93705571e+00 -3.36865291e+02] [3.17690088e-01 -8.72908094e-01
2.67370854e+02 1.74863348e+00 -1.93705571e+00 -3.36672197e+02]]

Crossover

[[3.17690088e-01 -8.72908094e-01 2.67689952e+02 1.74863348e+00 -
1.93705571e+00 - 3.36689592e+02]

[3.17690088e-01 -8.72908094e-01 2.67638232e+02 1.74863348e+00

-1.93705571e+00 -3.36865291e+02]

[3.17690088e-01 -8.72908094e-01 2.67254110e+021.74863348e+00

-1.93705571e+00 -3.36672197e+02]

[3.17690088e-01 -8.72908094e-01 2.67370854e+021.74863348e+00

-1.93705571e+00 -3.37108802e+02]]

Mutation

[[3.17690088e-01 -8.72908094e-01 2.68382875e+021.74863348e+00

-1.93705571e+00 -3.36222272e+02]

[3.17690088e-01 -8.72908094e-01 2.68456819e+021.74863348e+00

-1.93705571e+00 -3.37417363e+02]

[3.17690088e-01 -8.72908094e-01 2.67606746e+021.74863348e+00

-1.93705571e+00 -3.36866918e+02]

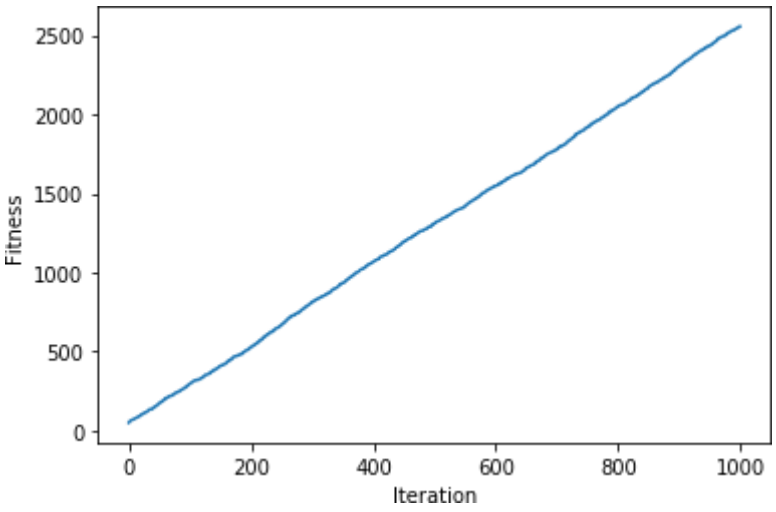
[3.17690088e-01 -8.72908094e-01 2.67051753e+021.74863348e+00

-1.93705571e+00 -3.37331663e+02]]

Best solution : [[[3.17690088e-01 -8.72908094e-01 2.68456819e+021.74863348e+00

-1.93705571e+00-
3.37417363e+02]]]

Best solution fitness
:[2558.52782726]



VIVA QUESTIONS & ANSWERS

1. What is a Genetic Algorithm?

Let's get back to the example we discussed above and summarize what we did.

1. Firstly, we defined our initial population as our countrymen.
2. We defined a function to classify whether is a person is good or bad.

3. Then we selected good people for mating to produce their off-springs.
4. And finally, these off-springs replace the bad people from the population and this process repeats.

PROGRAM 9

9.IMPLEMENT THE FINITE WORDS CLASSIFICATION SYSTEM USING BACK-PROPAGATIONALGORITHM

AIM:

To implement thefinite words classification system using Back-propagationalgorithm

DESCRIPTION:

What is backpropagation?

We can define the backpropagation algorithm as an algorithm that trains some given feed-forward Neural Network for a given input pattern where the classifications are known to us. At the point when every passage of the example set is exhibited to the network, the network looks at its yield reaction to the example input pattern. After that, the comparison done between output response and expected output with the error value is measured. Later, we adjust the connection weight based upon the error value measured.

It was first introduced in the 1960s and 30 years later it was popularized by David Rumelhart, Geoffrey Hinton, and Ronald Williams in the famous 1986 paper. In this paper, they spoke about the various neural networks. Today, back propagation is doing good. Neural network training happens through back propagation. By this approach, we fine-tune the weights of a neural net based on the error rate obtained in the previous run. The right manner of applying this technique reduces error rates and makes the model more reliable. Backpropagation is used to train the neural network of the chain rule method. In simple terms, after each feed-forward passes through a network, this algorithm does the backward

pass to adjust the model's parameters based on weights and biases. A typical supervised learning algorithm attempts to find a function that maps input data to the right output. Back propagation works with a multi-layered neural network and learns internal representations of input to output mapping.

The Back propagation algorithm is a supervised learning method for multilayer feed-forward networks from the field of Artificial Neural Networks.

Feed-forward neural networks are inspired by the information processing of one or more neural cells, called a neuron. A neuron accepts input signals via its dendrites, which pass the electrical signal down to the cell body. The axon carries the signal out to synapses, which are the connections of a cell's axon to other cell's dendrites.

The principle of the backpropagation approach is to model a given function by modifying internal weightings of input signals to produce an expected output signal. The system is trained using a supervised learning method, where the error between the system's output and a known expected output is presented to the system and used to modify its internal state.

Technically, the backpropagation algorithm is a method for training the weights in a multilayer feed-forward neural network. As such, it requires a network structure to be defined of one or more layers where one layer is fully connected to the next layer. A standard network structure is one input layer, one hidden layer, and one output layer.Back propagation can be used for both classification and regression problems.

In classification problems, best results are achieved when the network has one neuron in the output layer for each class value. For example, a 2-class or binary classification problem with the class values of A and B. These expected outputs would have to be transformed into binary vectors with one column for each class value. Such as [1, 0] and [0, 1] for A and B respectively. This is called a one hot encoding.

How does backpropagation work?

Let us take a look at how backpropagation works. It has four layers: input layer, hidden layer, hidden layer II and final output layer.

So, the main three layers are:

1. Input layer
2. Hidden layer
3. Output layer

Each layer has its own way of working and its own way to take action such that we are able to get the desired results and correlate these scenarios to our conditions. Let us discuss other details needed to help summarizing this algorithm.

SOURCE CODE:

```
import pandas as pd

fromsklearn.model_selection import train_test_split

fromsklearn.feature_extraction.text import

CountVectorizer from sklearn.neural_network

import MLPClassifier

fromsklearn.metrics import accuracy_score, confusion_matrix, precision_score, recall_score


msg = pd.read_csv('document.csv',

names=['message', 'label']) print("Total Instances of

Dataset: ", msg.shape[0]) msg['labelnum'] =

msg.label.map({'pos': 1, 'neg': 0})


X =

msg.me

ssage y
```

```
=

msg.lab

elnum

Xtrain, Xtest, ytrain, ytest = train_test_split(X, y)

count_v = CountVectorizer()

Xtrain_dm =

count_v.fit_transform(Xtrain)

Xtest_dm =

count_v.transform(Xtest)

df = pd.DataFrame(Xtrain_dm.toarray(),columns=count_v.get_feature_names())

clf = MLPClassifier(solver='lbfgs', alpha=1e-5,hidden_layer_sizes=(5, 2), random_state=1)

clf.fit(Xtrain_dm, ytrain) pred = clf.predict(Xtest_dm)

print('Accuracy Metrics:')

   print('Accuracy: ', accuracy_score(ytest, pred)) print('Recall: ', recall_score(ytest, pred))
print('Precision: ',        precision_score(ytest, pred))

print('Confusion Matrix: \n', confusion_matrix(ytest, pred))
```

document.csv:

```
I love this

sandwich,pos This

is an

amazingplace,pos
```

I feel very good about these

beers,pos This is my best

work,pos

What an awesome view,pos

I do not like this

restaurant,neg I am

tired of this stuff,neg

I can't deal with

this,neg He is

my sworn

enemy,neg My

boss is

horrible,neg

This is an awesome place,pos

I do not like the taste of

this juice,neg I love to

dance,pos

I am sick and tired of this

place,neg What a great

holiday,pos

That is a bad locality to stay,neg
We will have good fun

tomorrow,pos I went to my

enemy's house today,neg

OUTPUT:

Total Instances of

Dataset: 18

Accuracy Metrics:

Accuracy: 0.8

Recall: 1.0

Precisio

n: 0.75

Confusi

on

Matrix:

[[1 1]

[0 3]

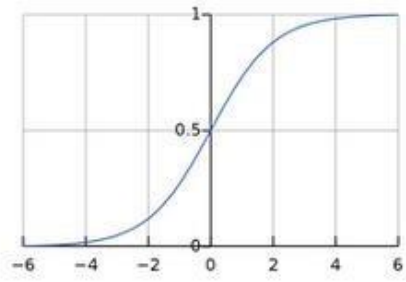
VIVA QUESTIONS & ANSWERS:

1. What is the difference between a Perceptron and Logistic Regression?

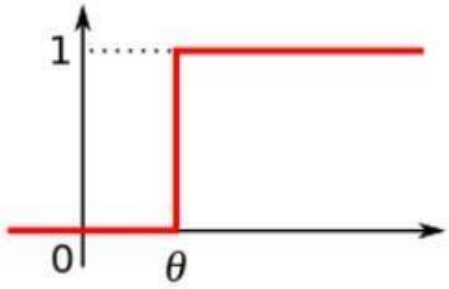
A Multi-Layer Perception (MLP) is one of the most basic neuralnetworks that we use for classification. For a binary classification problem, we know that the output can be either 0 or 1. This is just like our simple logistic regression, where we use a logit function to generate a probability between 0 and 1.

So, what's the difference between the two?

Simply put, it is just the difference in the threshold function! When we restrict the logistic regression model to give us either exactly 1 or exactly 0, we get a Perception model:



Logistic regression - Logit function



Perceptron step function

2. Can we have the same bias for all neurons of a hidden layer?

Essentially, you can have a different bias value at each layer or at each neuron as well. However, it is best if we have a bias matrix for all the neurons in the hidden layers as well.

A point to note is that both these strategies would give you very different results.

3. What if we do not use any activation function(s) in a neural network?

The main aim of this question is to understand why we need activation functions in a neural network. You can start off by giving a simple explanation of how neural networks are built:

Step 1: Calculate the sum of all the inputs (X) according to their weights and include the bias term:

Z = (weights * X) + bias

Step 2: Apply an activation function to calculate the expected output:

Y = Activation(Z)

Steps 1 and 2 are performed at each layer. If you recollect, this is nothing but forward propagation! Now, what if there is no activation function?

Our equation for Y essentially becomes:

Y = Z = (weights * X) + bias

Wait – isn't this just a simple linear equation? Yes – and that is why we need activation functions. A linear equation will not be able to capture the complex patterns in the data – this is even more evident in the case of deep learning problems.

In order to capture non-linear relationships, we use activation functions, and that is why a neural network without an activation function is just a linear regression model.

4. In a neural network, what if all the weights are initialized with the same value?

In simplest terms, if all the neurons have the same value of weights, each hidden unit will get exactly the same signal. While this might work during forward propagation, the derivative of the cost function during backward propagation would be the same every time.

In short, there is no learning happening by the network! What do you call the phenomenon of the model being unable to learn any patterns from the data? Yes, under fitting.

Therefore, if all weights have the same initial value, this would lead to under fitting.

5. What is the role of weights and bias in a neural network?

This is a question best explained with a real-life example. Consider that you want to go out today to play a cricket match with your friends. Now, a number of factors can affect your decision-making, like:

- How many of your friends can make it to the game?
- How much equipment can all of you bring?
- What is the temperature outside?

And so on. These factors can change your decision greatly or not too much. For example, if it is raining outside, then you cannot go out to play at all. Or if you have only one bat, you can share it while playing as well. The magnitude by which these factors can affect the game is called the weight of that factor.

Factors like the weather or temperature might have a higher weight, and other factors like equipment would have a lower weight.

However, does this mean that we can play a cricket match with only one bat? No – we would need 1 ball and 6 wickets as well. This is where bias comes into the picture. Bias lets you assign some threshold which helps you activate a decision-point (or a neuron) only when that threshold is crossed.