## APPLICATION OF PROBABILITY

#### A PROJECT REPORT SUBMITTED

###### CH.SC.U4AIE23014 FAIZ RAHMAN

**CH.SC.U4AIE23018 GOWTHAM HARI**

###### CH.SC.U4AIE23019 HARIENDRAN T

**CH.SC.U4AIE23043 DHARSHINI R**

###### CH.SC.U4AIE23062 KOUSHIK V

Under the guidance of

**Dr . Ishwariya**

In partial fulfilment for the award of the degree of Bachelor Of Technology

In

##### Mathematics In Computing-1



AMRITA SCHOOL OF COMPUTING, CHENNAI AMRITA VISHWA VIDYAPEETHAM

CHENNAI-601103 DECEMBER-2023

##### AMRITA VISHWA VIDYAPEETHAM

**AMRITA SCHOOL OF ENGINEERING, CHENNAI-**

##### 601103



**BONAFIDE CERTIFICATE**

This is to certify that the major project report entitled “**MATHEMATICS FOR COMPUTING-1”**

##### Submitted by,

**CH.SC.U4AIE23014 FAIZ RAHMAN**

**CH.SC.U4AIE23018 GOWTHAM HARI**

**CH.SC.U4AIE23019 HARIENDRAN T**

**CH.SC.U4AIE23043 DHARSHINI R**

**CH.SC.U4AIE23062** **KOUSHIK .V**

In partial fulfilment of the requirements for the award of the Degree of Bachelor of Technology in **MATHEMATICS FOR COMPUTING-I** is a Bonafide record of the work carried out under my guidance and supervision at Amrita School of Engineering, Chennai.

Signature

##### Dr. Ishwariya

Asst.Professor, Department of Mathematics. This project was evaluated by us on …………

INTERNAL EXAMINER EXTERNAL EXAMINER

APPLICATION OF PROBABILITY

Faiz Rahman, Gowtham Hari, Hariendran T, Dharshini R, Koushik.V

**ABSTRACT**

There is a lot of uncertainty in the world, but agents still need to act. Probability provides a way of summarizing the uncertainty that comes from our laziness and ignorance. The theory of probability deals with averages of mass phenomena occurring sequentially or simultaneously: electron emission, telephone calls, radar detection, quality control, system failure, games of chance, statistical mechanics, turbulence, noise, birth and death rates, and queueing theory, among many others. In this report we will be studying the use of probability in Artificial Intelligence, various probability rules.

We will also get deeper into the Bayes Theorem, Bayesian Statistics, Monte Carlo Simulation and their applications. Bayes' theorem is an important part of inference statistics and many advanced machine learning models. Bayesian inference is a logical approach to updating the potential of hypotheses in the light of new knowledge, and therefore naturally plays a central role in science. It explains the likelihood of an event based on prior knowledge of circumstances that may be relevant to the event. Bayes' theorem provides a method of calculating the degree of uncertainty. (Berrar, 2018). It can be applied in our daily lives when we are attempting to make a decision based on new information. The aim of this research is to shed light on the various fields in which this theory is applied.

Monte Carlo (MC) approach to analysis was developed in the 1940's, it is a computer based analytical method which employs statistical sampling techniques for obtaining a probabilistic approximation to the solution of a mathematical equation or model by utilizing sequences of random numbers as inputs into a model which yields results that are indications of the performance of the developed model.

Keywords: Probability, Artificial Intelligence, Monté Carlo Simulation, Bayes Theorem, Bayesian Statistics.



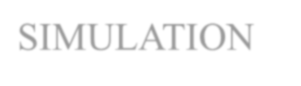
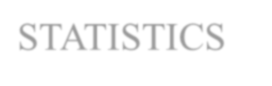


TABLE OF CONTENTS

|  |  |  |
| --- | --- | --- |
| Chapter | Title | Page no |
| 1. | ABSTRACT | 3 |
| 2. | INTRODUCTION | 5 |
| 3. | LITERATURE REVIEW | 6 |
| 4. | PROBABILITY | 7 |
| 5. | HISTORY OF PROBABILITY | 8 |
| 6. | UNCERTAINTY | 10 |
| 7. | BAYES’ THEOREM | 11 |
| 8. | BAYESIAN BELIEFS NETWORK | 25 |
| 9. | MONTE CARLO SIMULATION | 35 |
| 10. | CONCLUSION | 52 |
| 11. | REFERENCE | 53 |

**INTRODUCTION:**

Probability is the branch of mathematics that studies the possible outcomes of the given events together with the outcomes relative likelihoods and distributions. In common usage, the word “probability” is used to mean the chance that a particular event (or set of events) will occur expressed on a linear scale from 0 (impossibility) to 1 (certainty), also expressed as a percentage between 0 and 100%. One of the key concepts in probability is the probability space, which consists of a sample space, an event space, and a probability measure. The sample space represents all possible outcomes of a random experiment, while the event space consists of subsets of the sample space corresponding to specific events. The probability measure assigns a numerical value between 0 and 1 to each event, with 0 indicating impossibility and 1 indicating certainty. The analysis of events governed by probability is called statistics. Thus, probability theory makes the same ontological commitment as logic namely, that facts either do or do not hold in the world.

Probability is a fundamental concept that plays a crucial role in various fields such as mathematics, statistics, physics, finance, and beyond. At its core, probability is the measure of the likelihood that a particular event will occur. Whether it's predicting the outcome of a coin toss, understanding the chances of winning a game, or assessing the risk in financial investments, probability provides a framework for making informed decisions in the face of uncertainty.

Preferences, as expressed by utilities, are combined with probabilities in the general theory of rational decisions called decision theory. **Decision theory= probability theory + utility theory**.

The fundamental idea of decision theory is that an agent is rational if and only if it chooses the action that yields the highest expected utility, averaging over all the possible outcomes of the action. This is called the principle of Maximum Expected Utility (MEU).

**For example**, when we toss a coin, either we get Head OR Tail, only two possible outcomes are possible (H, T). But when two coins are tossed then there will be four possible outcomes, i.e {(H, H), (H, T), (T, H), (T, T)}.

The probability formula is defined as the possibility of an event to happen is equal to the ratio of the number of favourable outcomes and the total number of outcomes.

|  |
| --- |
| **Probability of event to happen P(E) = Number of favourable outcomes/Total Number of outcomes** |

Sometimes students get mistaken for “favourable outcome” with “desirable outcome”. This is the basic formula. But there are some more formulas for different situations or events.

LITERATURE REVIEW:

* The paper proposed by Sudipta Das, Girija Nandan Kar of mainly focuses on the use of Probability in Artificial Intelligence.
* The paper proposed by Alexander Suhobokov deals with Monte Carlo simulation method and its application in Risk Management.
* Foundations of Probability Theory for AI by Michael Kearns and Michael Jordan provides a comprehensive introduction to probability theory and its applications in AI.
* Algorithmic Probability and Causal Inference by Judea Pearl and Dana Mackenzie
* dives deep into the intersection of probability, statistics, and causal reasoning, crucial for AI that makes decisions in the real world.
* Probabilistic Graphical Models: A Bayesian Tutorial by Koller and Friedman (2009): This widely used resource explains graphical models, a powerful tool for representing and reasoning about probabilistic relationships.
* Deep Learning with Bayesian Principles and Uncertainty Quantification by Gal et al. (2016): This paper explores how Bayesian techniques can be used to improve the accuracy and reliability of deep learning models.
* Learning Probabilistic Inference Networks with Local Likelihood Constraints by Noh et al. (2017): This research introduces a new approach for training probabilistic models that are more robust to noisy data.

#### PROBABILITY:

Probability is a measure of the likelihood of an event occurring. It is expressed as a number between 0 and 1, with 0 representing an impossible event and 1 representing a certain event.

Probability is used to help us understand the world around us, from predicting the outcome of a coin toss to estimating the probability of a disease outbreak.

There are two types of Probability: classical and empirical probability. Classical probability is based on theoretical considerations and assumes that all outcomes in a sample space are equally likely. For example, when rolling a fair six-sided die, each of the six faces has an equal chance of landing face up, making classical probability applicable.

Empirical probability, on the other hand, relies on observed data and frequencies. It involves conducting experiments or collecting real-world data to estimate the likelihood of events.

This type of probability is particularly useful in situations where theoretical assumptions may not hold, and outcomes are influenced by practical factors.

Probability theory is also divided into two main branches: discrete and continuous probability. Discrete probability deals with situations where the set of possible outcomes is countable, such as the number of heads in multiple coin tosses. Continuous probability, on the other hand, addresses scenarios with an infinite number of possible outcomes within a given range, such as the height of individuals in a population.

## **Applications of Probability**

Probability has a wide variety of applications in real life. Some of the common applications which we see in our everyday life while checking the results of the following events:

* Choosing a card from the deck of cards
* Flipping a coin
* Throwing a dice in the air
* Pulling a red ball out of a bucket of red and white balls
* Winning a lucky draw

### Other Major Applications of Probability

* It is used for risk assessment and modelling in various industries
* Weather forecasting or prediction of weather changes
* Probability of a team winning in a sport based on players and strength of team
* In the share market, chances of getting the hike of share prices

HISTORY OF PROBABILITY:

* 17th century records the first documented evidence of the study of probability. More precisely, in 1654, a French scientist, Chevalier de Mere studied questions related to gambling.
* The above problem called the attention of the great mathematician Blaise Pascal. The same problem led to the exchange of letters between Pascal and another renowned mathematician Pierre de Fermat.
* This exchange of letters is termed as the famous Pascal Fermat Correspondence in the history of probability.
* The major contributors to probability in 17th and 18th century were Jacob Bernoulli and Abraham de Moivre. They worked on the mathematical formulation of the theory.
* 19th century was marked by the work of Laplace. Some other significant contributors were Chebyshev, Markov and Kolmogorov.
* The famous axiomatic definition of probability was formulated by Kolmogorov in 1933.

## **Probability of an Event**

Assume an event E can occur in r ways out of a sum of n probable or possible **equally likely ways**. Then the probability of happening of the event or its success is expressed as:

P(E) = r/n

The probability that the event will not occur or known as its failure is expressed as:

P(E’) = (n-r)/n = 1-(r/n)

E’ represents that the event will not occur.

Therefore, now we can say;

**P(E) + P(E’) = 1**

This means that the total of all the probabilities in any random test or experiment is equal to 1.

### What are Equally Likely Events?

When the events have the same theoretical probability of happening, then they are called equally likely events. The results of a sample space are called equally likely if all of them have the same probability of occurring. For example, if you throw a die, then the probability of getting 1 is 1/6. Similarly, the probability of getting all the numbers from 2,3,4,5 and 6, one at a time is 1/6. Hence, the following are some examples of equally likely events when throwing a die:

* Getting 3 and 5 on throwing a die
* Getting an even number and an odd number on a die
* Getting 1, 2 or 3 on rolling a die

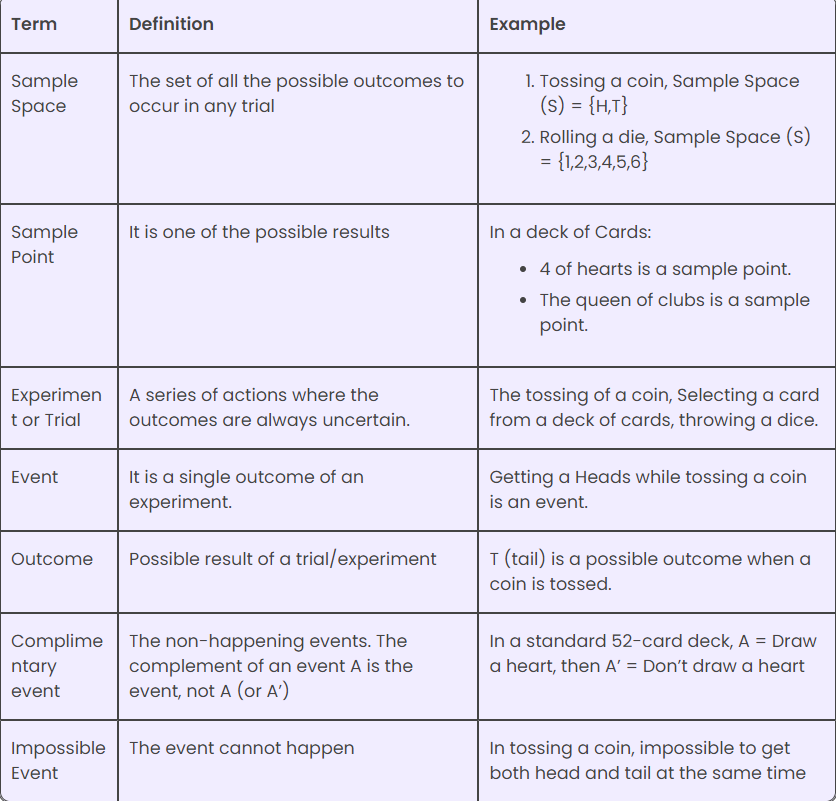
are equally likely events, since the probabilities of each event are equal.

### Complementary Events

The possibility that there will be only two outcomes which states that an event will occur or not. Like a person will come or not come to your house, getting a job or not getting a job, etc. are examples of complementary events. Basically, the complement of an event occurring in the exact opposite that the probability of it is not occurring. Some more examples are:

* It will rain or not rain today
* The student will pass the exam or not pass.
* You win the lottery or you don’t.

## **Probability Terms and Definition**



**UNCERTAINITY**

**Uncertainty**

From self-driving cars to virtual personal assistants, AI technologies have become integral to our daily routines. However, one of the key challenges that AI systems face is dealing with uncertainty.

Uncertainty arises due to various factors such as unreliable sources of Information, experimental errors, equipment faults, temperature variations, and climate change, among others. To address this challenge, probabilistic reasoning techniques have gained significant importance in AI, allowing machines to make decisions and predictions in uncertainty.

**Causes of Uncertainty**

Uncertainty in AI can arise from various sources, including:

Information Occurred from Unreliable Sources:

AI systems rely on data to make decisions and predictions. However, data obtained from various sources may not always be reliable. Data can be incomplete, inconsistent, or biased, leading to uncertainty in the outcomes generated by AI systems.

* Experimental Errors:

In scientific research and experimentation, errors can occur at various stages, such as data collection, measurement, and analysis.

These errors can introduce uncertainty in the results and conclusions drawn from the experiments.

* Equipment Fault:

In many AI systems, machines and sensors are used to collect data and make decisions. However, these machines can be subject to faults, malfunctions, or inaccuracies, leading to uncertainty in the outcomes generated by AI systems.

* Temperature Variation:

Many real-world applications of AI, such as weather prediction, environmental monitoring, and energy management, are sensitive to temperature variations.

However, temperature measurements can be subject to uncertainty due to factors such as sensor accuracy, calibration errors, and environmental fluctuations.

* Climate Change:

Climate change is a global phenomenon that introduces uncertainty in various aspects of our lives. For example, predicting the impacts of climate change on agriculture, water resources, and infrastructure requires dealing with uncertain data and models

### Probabilistic Reasoning

Probabilistic reasoning is a technique used in AI to address uncertainty by modelling and reasoning with probabilistic information. It allows AI systems to make decisions and predictions based on the probabilities of different outcomes, taking into account uncertain or incomplete Information. Probabilistic reasoning provides a principled approach to handling uncertainty, allowing machines to reason about uncertain situations in a rigorous and quantitative manner.

### Need for Probabilistic Reasoning in AI

The need for probabilistic reasoning in AI arises because uncertainty is inherent in many real- world applications.

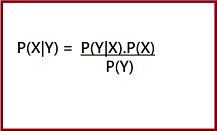
**For example,** there is often uncertainty in the symptoms, test results, and patient history in medical diagnosis. In autonomous vehicles, there is uncertainty in the sensor measurements, road conditions, and traffic patterns. In financial markets, there is uncertainty in stock prices, economic indicators, and investor behaviour. Probabilistic reasoning techniques allow AI systems to deal with these uncertainties and make informed decisions.

**There Are Two Ways to Solve Problems with Uncertain Knowledge:**

### Bayes’ Theorem:

Bayes' theorem is also known as Bayes' rule, Bayes' law, or Bayesian reasoning, which determines the probability of an event with uncertain knowledge. In probability theory, it relates the conditional probability and marginal probabilities of two random events. Bayes' theorem was named after the British mathematician Thomas Bayes. The Bayesian inference is an application of Bayes' theorem, which is fundamental to Bayesian statistics.

Mathematically, Bayes' Theorem is expressed as:



Here, both events X and Y are independent events which means probability of outcome of both events does not depends one another.

Where:

* **P(A|B)** represents the posterior probability, which is the probability of event A occurring given that event B has occurred.
* **P(B|A)** represents the likelihood, which is the probability of observing event B given that event A has occurred.
* **P(A)** represents the prior probability, which is the initial probability of event A occurring before considering any new evidence.
* **P(B)** represents the marginal likelihood, which is the probability of observing event B, regardless of whether event A has occurred.
  + Hence, Bayes Theorem can be written as:

**posterior = likelihood \* prior / evidence.**

In the context of AI, Bayes' Theorem is used to update the probabilities of different hypotheses or predictions based on new data or evidence. It is particularly useful in handling uncertainty and making decisions when there is incomplete or ambiguous Information.

### Prerequisites for Bayes’ Theorem

While studying the Bayes’ theorem, we need to understand few important concepts. These are as follows:

1. Experiment

An experiment is defined as the planned operation carried out under controlled condition such as tossing a coin, drawing a card and rolling a dice, etc.

1. Sample Space

During an experiment what we get as a result is called as possible outcomes and the set of all possible outcome of an event is known as sample space. For example, if we are rolling a dice, sample space will be:

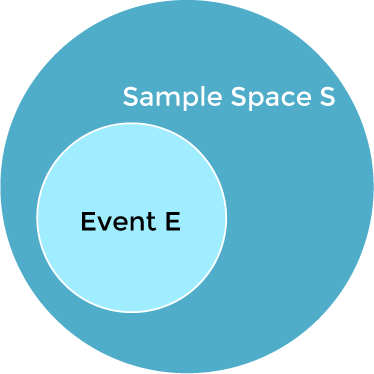
S1 = {1, 2, 3, 4, 5, 6}

Similarly, if our experiment is related to toss a coin and recording its outcomes, then sample space will be:

S2 = {Head, Tail}

1. Event

Event is defined as subset of sample space in an experiment. Further, it is also called as set of outcomes.



Assume in our experiment of rolling a dice, there are two event A and B such that; A = Event when an even number is obtained = {2, 4, 6}

B = Event when a number is greater than 4 = {5, 6}

* **Probability of the event A ''P(A)''**= Number of favourable outcomes / Total number of possible outcomes

P(E) = 3/6 =1/2 =0.5

* Similarly, **Probability of the event B ''P(B)''**= Number of favourable outcomes / Total number of possible outcomes

=2/6

=1/3

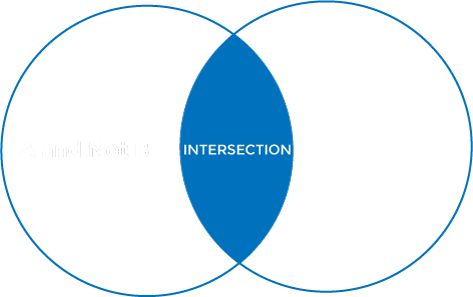
=0.333

* Union of event A and B:

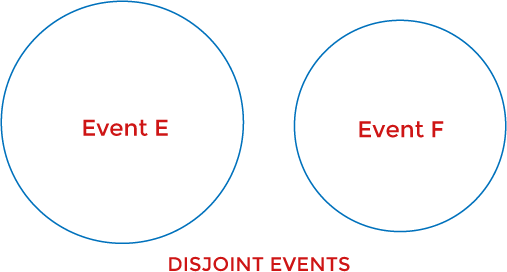
A𝖴B = {2, 4, 5, 6}

* **Intersection of event A and B:**

A∩B= {6}

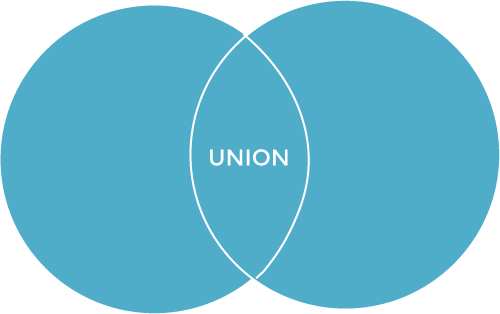


* **Disjoint Event:** If the intersection of the event A and B is an empty set or null then such events are known as **disjoint event** or **mutually exclusive events** also.



1. Random Variable:

It is a real value function which helps mapping between sample space and a real line of an



experiment. A random variable is taken on some random values and each value having some probability. However, it is neither random nor a variable, but it behaves as a function which can either be discrete, continuous or combination of both.

1. Exhaustive Event:

As per the name suggests, a set of events where at least one event occurs at a time, called exhaustive event of an experiment. Thus, two events A and B are said to be exhaustive if

either A or B occur at a time and both are mutually exclusive for e.g., while tossing a coin, either it will be a Head or may be a Tail.

1. Independent Event:

Two events are said to be independent when occurrence of one event does not affect the occurrence of another event. In simple words we can say that the probability of outcome of both events does not depends one another.

Mathematically, two events A and B are said to be independent if: P (A ∩ B) = P(AB) = P(A)\*P(B)

1. Conditional Probability:

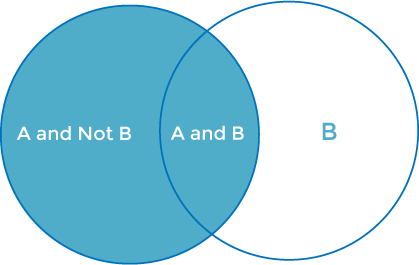
Conditional probability is defined as the probability of an event A, given that another event B has already occurred (i.e., A conditional B). This is represented by P(A|B) and we can define it as:

P(A|B) = P (A ∩ B) / P(B)

1. Marginal Probability:

Marginal probability is defined as the probability of an event A occurring independent of any other event B. Further, it is considered as the probability of evidence under any consideration.

P(A) = P(A|B) \*P(B) + P(A|~B) \*P(~B)



# Application of Bayes theorem:

###### 1.AUTONOMOUS VEHICLES: -

The role of Bayes Theorem in error detection and correction:

Bayes Theorem plays an important role in the error detection and correction process in autonomous vehicles. Bayes’ Theorem can be used to detect and correct sensor errors in the following ways:

Accurate Risk Assessment: Bayes Theorem helps to make an accurate risk assessment in autonomous vehicles. More accurate risk estimates can be made by integrating previously known information and up-to-date data. This increases the safety of vehicles and the confidence of users.

Error Detection and Correction: Bayes Theorem plays an effective role in detecting and correcting sensor errors. By analysing sensor data and using previously known information, erroneous measurements can be corrected, and accurate results can be obtained. This enables autonomous vehicles to make more reliable and accurate decisions.

Improving the Learning Process: Bayes’ Theorem can be used effectively in the learning process of autonomous vehicles. With the integration of data collection, analysis and machine learning techniques, the tools better understand their environment and predict future situations. This improves the performance of the tools and increases the user experience.

Perception:

Autonomous vehicles use various sensors like cameras, lidar, radar, and GPS to perceive the environment. Bayes' Theorem can be applied to fuse information from these sensors, adjusting the probability of the vehicle's actual state given sensor observations.

Object Recognition:

When the car needs to recognize objects in its surroundings (pedestrians, other vehicles, etc.), Bayes' Theorem can be applied to update the belief about the identity of an object based on sensor data.

Decision-Making:

Bayesian decision theory is used for decision-making in uncertain environments. The vehicle can update its decisions based on incoming data, adjusting probabilities of different outcomes.

Localization:

Bayes' Theorem is often used in localization algorithms. Given sensor measurements and a map of the environment, the algorithm can estimate the probability distribution of the vehicle's location.

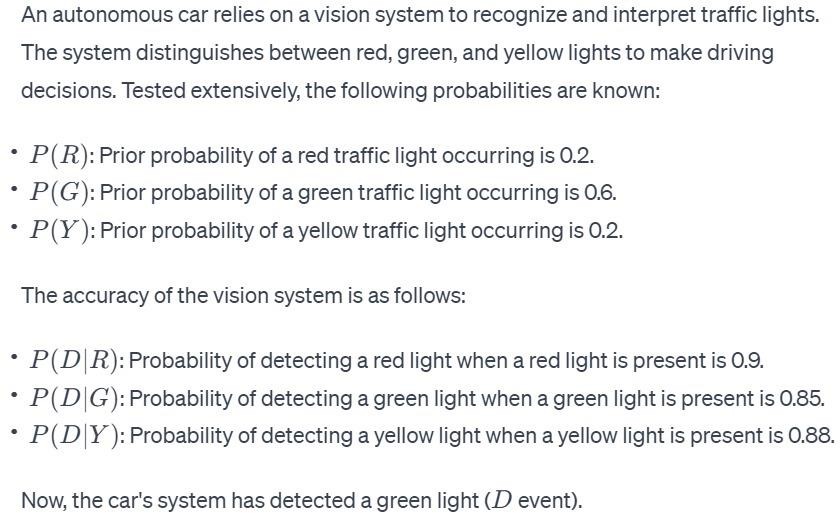
Sensor Fusion:

Autonomous vehicles often rely on multiple sensors to get a comprehensive view of the environment. Bayesian methods are applied to fuse information from different sensors, providing a more accurate representation of the surroundings.

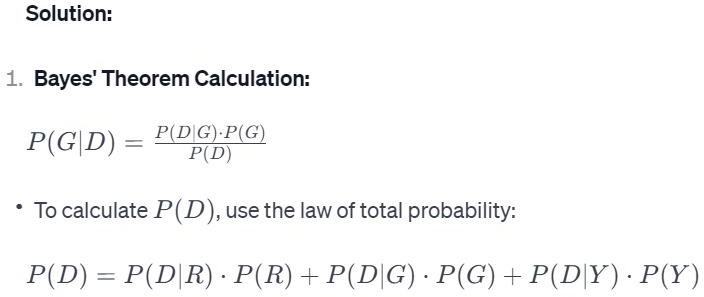
**BAYESIAN OPTIMIZATION**

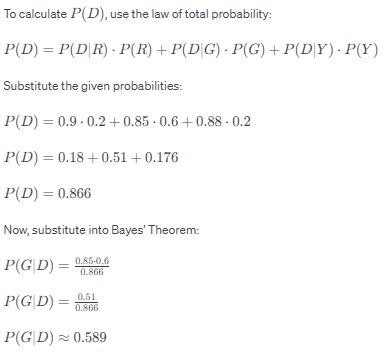
This technique is mainly used for optimizing black box functions. Simply put, this requires changing the parameters of the function until an optimal solution is achieved based on previously observed function evaluations.

###### Traffic Light Recognition

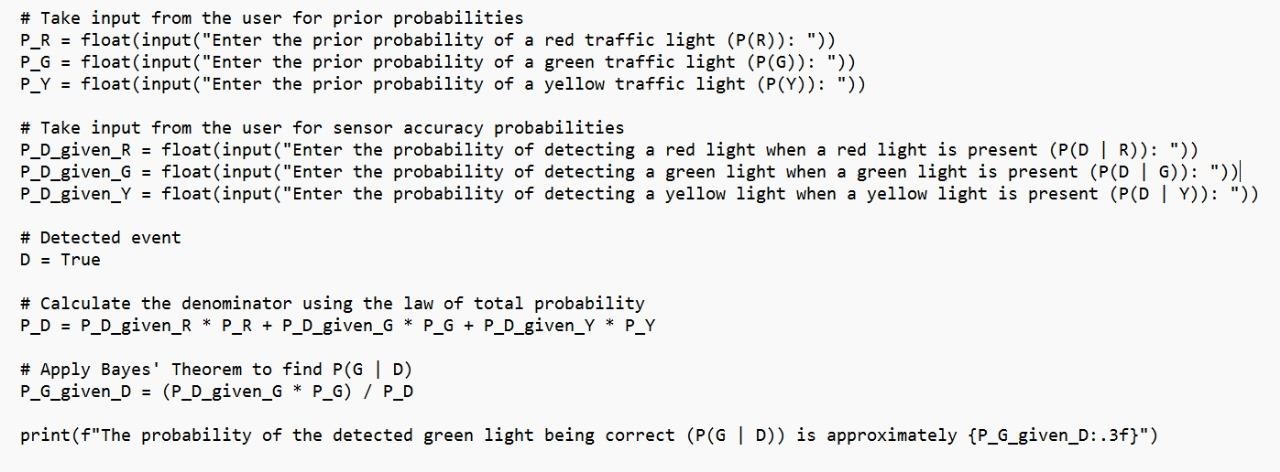


Calculate the probability that the detected green light is indeed a green light *P*(*G*∣*D*)).





AUTONOMOUS VEHICLE PROBLEM USING BAYES THEOREM: - TRAFFIC LIGHT DETECTION:



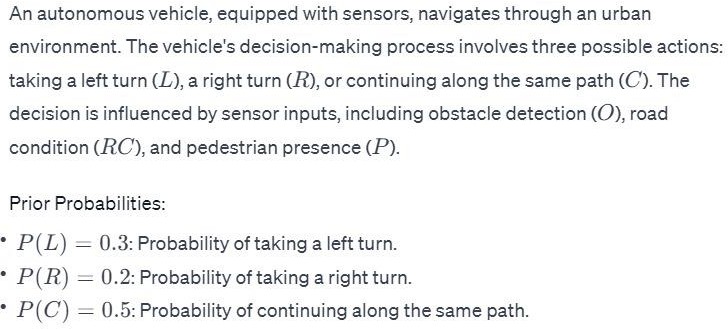
**OUTPUT**

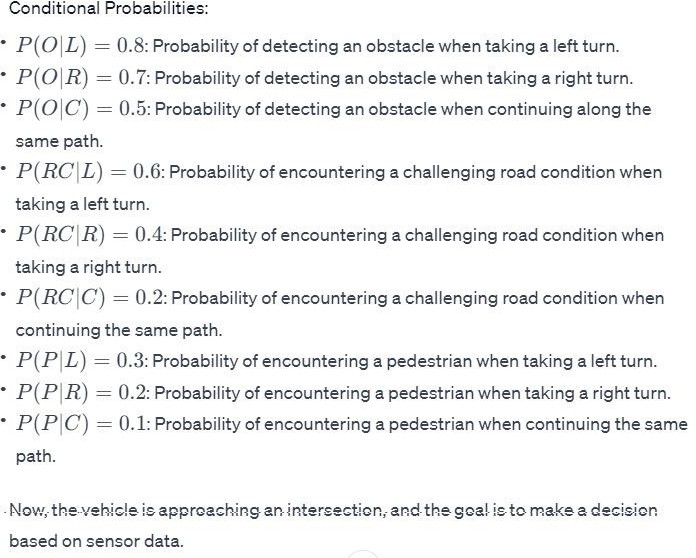
Enter the prior probability of a red traffic light (P(R)): .2 Enter the prior probability of a green traffic light (P(G)): .6 Enter the prior probability of a yellow traffic light (P(Y)): .2

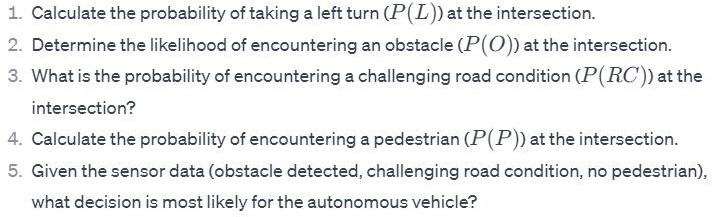
Enter the probability of detecting a red light when a red light is present (P (D | R)): .9

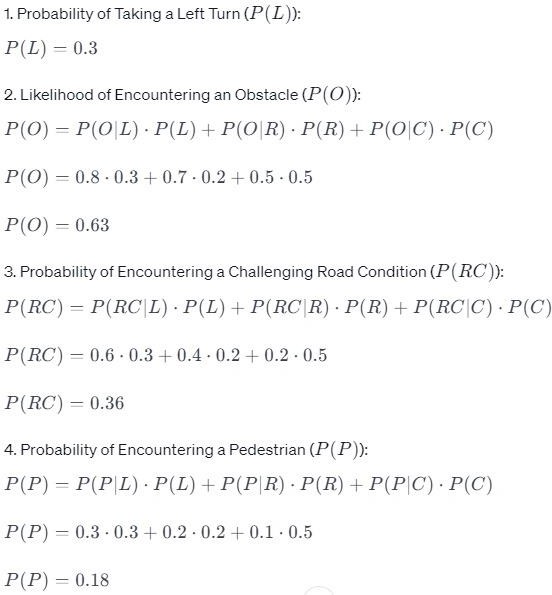
Enter the probability of detecting a green light when a green light is present (P (D | G)): .85 Enter the probability of detecting a yellow light when a yellow light is present (P (D | Y)): .88 The probability of the detected green light being correct (P (G | D)) is approximately 0.589

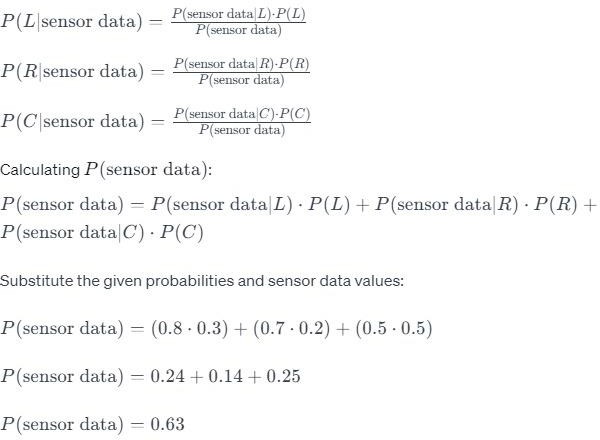
**Decision-Making in Autonomous Navigation:**

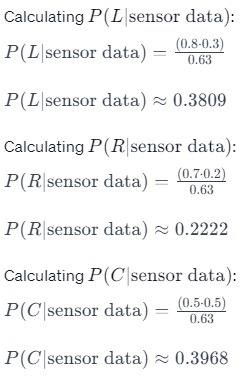




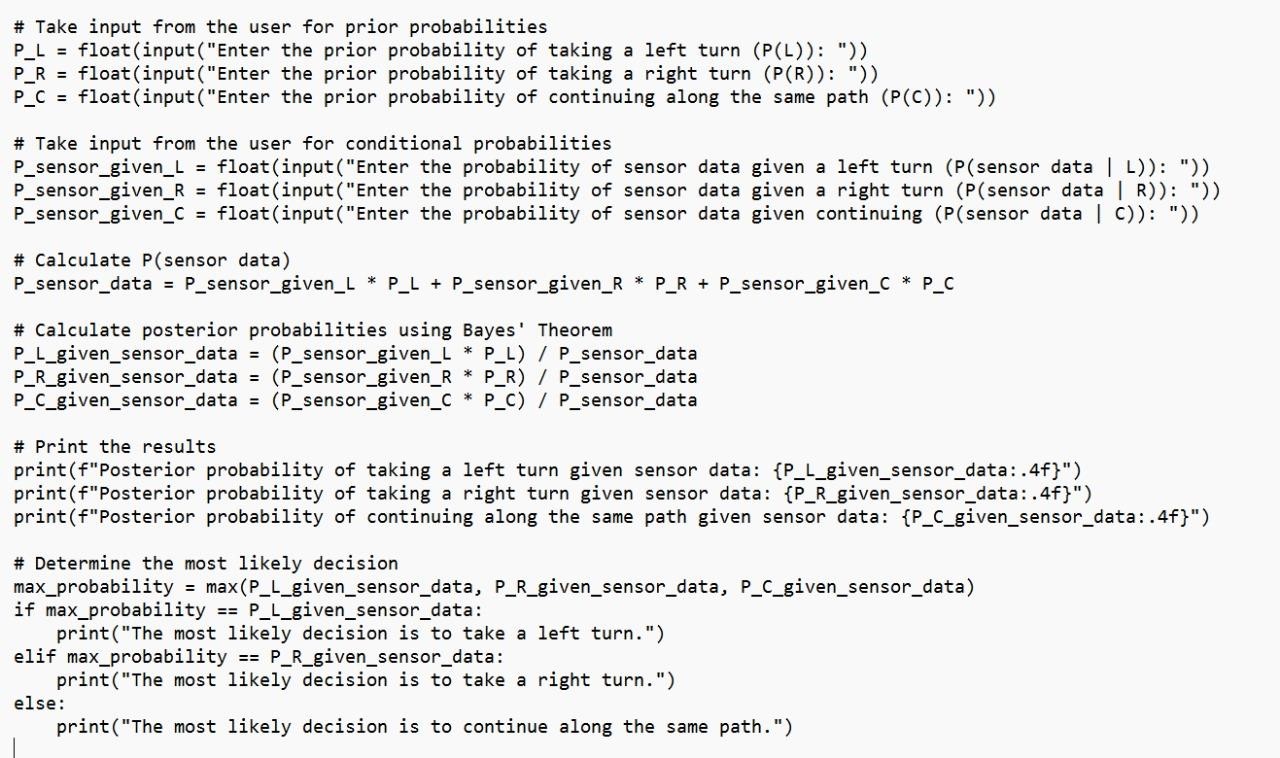


**SOLUTION:**





**PYTHON CODE FOR DECIDING PATH :**



OUTPUT:

Enter the prior probability of taking a left turn (P(L)): .4 Enter the prior probability of taking a right turn (P(R)): .5

Enter the prior probability of continuing along the same path (P(C)): .6 Enter the probability of sensor data given a left turn (P (sensor data | L)): .66

Enter the probability of sensor data given a right turn (P (sensor data | R)): .77 Enter the probability of sensor data given continuing (P (sensor data | C)): .88 Posterior probability of taking a left turn given sensor data: 0.2243

Posterior probability of taking a right turn given sensor data: 0.3271

Posterior probability of continuing along the same path given sensor data: 0.4486 The most likely decision is to continue along the same path.

**OTHER APPLICATIONS OF BAYES’ THEOREM:**

* Medical diagnosis: Bayes' theorem is used to calculate the probability of a disease given certain symptoms. This can help doctors make more accurate diagnoses and provide better treatment. For example, Bayes' theorem can be used to calculate the probability of a patient having a heart attack given their symptoms, such as chest pain and shortness of breath.
* Spam filtering: Bayes' theorem is used in spam filters to identify spam emails. Spam filters work by calculating the probability that an email is spam given certain features, such as the sender's address, the subject line, and the words in the body of the email.
* Weather forecasting: Bayes' theorem is used in weather forecasting to update the probability of different weather conditions based on new data, such as measurements from weather stations and satellite imagery. For example, Bayes' theorem can be used to calculate the probability of rain tomorrow given the current temperature, humidity, and wind speed.
* Fraud detection: Bayes' theorem is used in fraud detection systems to identify fraudulent activity. Fraud detection systems work by calculating the probability that an activity is fraudulent given certain features, such as the type of transaction, the amount of the transaction, and the location of the transaction.
* Recommendation systems: Bayes' theorem is used in recommendation systems to recommend products or services to users. Recommendation systems work by calculating the probability that a user will like a product or service given the user's past behavior and preferences.
* Image and video recognition: Bayes' theorem is used in image and video recognition to identify objects, people, and scenes in images and videos. Image and video recognition algorithms work by calculating the probability of different possible interpretations of an image or video given the pixels in the image or video.
* Robotics: Bayes' theorem is used in robotics to help robots make decisions about how to move and interact with their environment. Robots work by calculating the probability of different possible outcomes given their sensors and actuators.
* Games: Bayes' theorem is used in games to create intelligent opponents. Game AI algorithms work by calculating the probability of different possible moves or strategies given the current state of the game.

### Bayesian Belief Network

### Introduction

In the field of artificial intelligence and decision-making, Bayesian Belief Networks (BBNs) have emerged as a powerful tool for probabilistic reasoning and inference. BBNs provide a framework for representing and analysing complex systems by explicitly modelling the relationships between uncertain variables. With their ability to reason under uncertainty, BBNs have found wide-ranging applications in areas such as healthcare, finance, environmental management, and more. In this technical article, we will explore the fundamentals of Bayesian Belief Networks, their construction, inference algorithms, and real- world applications. Whether you are a researcher, practitioner or enthusiast in the field of AI, this article will provide you with a comprehensive understanding of BBNs and their potential for solving real-world problems.

1. Understanding Uncertainty:

In the real world, things are often uncertain. BBNs are tools that AI uses to handle this uncertainty. They're like virtual detectives, figuring out the best guesses based on the available clues.

1. Graphical Family Trees:

BBNs use cool graphical structures that look like family trees. Instead of representing relatives, though, these trees show how different pieces of information are related. Each piece of info is a 'node,' and the lines connecting them are 'edges.'

BBNs also use something called Bayes' theorem, which is like a wizard's spell for calculating probabilities. Each node in the network has its own

1. Conditional Probability Tables (CPTs):

These are like cheat sheets for each node. They tell you the probability of an event happening based on the conditions or evidence you have. Imagine predicting rain based on dark clouds

— that's a conditional probability!

1. Adapting to New Information:

The beauty of BBNs is that they're not stuck in their ways. As new information comes in, the network updates its beliefs. It's like adjusting your guess about whether your friend will join you for a movie as you get more texts from them.

1. Real-World Superpowers:

BBNs are superheroes in various fields. In medicine, they help diagnose diseases considering various symptoms. In finance, they assess risks based on market conditions. Wherever there's uncertainty, BBNs step in to bring order.

1. Challenges and Future Adventures:

BBNs have their limitations and face tricky situations. Researchers are always exploring ways to make them even more powerful and versatile.

Computing with Probabilities:

Law of Total Probability

P(a) = Σb P (a, b)

= Σb P (a | b) P(b) where B is any random variable

Why is this useful?

given a joint distribution (e.g., P(a,b,c,d)) we can obtain any “marginal”

probability (e.g., P(b)) by summing out the other variables, e.g.,

P(b) = Σa Σc Σd P (a, b, c, d)

Less obvious: we can also compute any conditional probability of interest given a

joint distribution, e.g.,

P (c | b) = Σa Σd P (a, c, d | b)

= (1 / P(b)) Σa Σd P (a, c, d, b)

where (1 / P(b)) is just a normalization constant

Thus, the joint distribution contains the information we need to compute any

probability of interest.

**Computing with Probabilities:**

The Chain Rule or Factoring We can always write

P (a, b, c, … z) = P (a | b, c, …. z) P (b, c, … z) (by definition of joint probability)

Repeatedly applying this idea, we can write P (a, b, c, … z) = P (a | b, c, …. z) P (b | c... z) P (c| ... z). P(z)

This factorization holds for any ordering of the variables This is the chain rule for probabilities

**Conditional Independence:**

* 2 random variables A and B are conditionally independent given C if P (a, b | c) = P (a | c) P (b | c) for all values a, b, c
* More intuitive (equivalent) conditional formulation
* A and B are conditionally independent given C iff

P (a | b, c) = P (a | c) OR P (b | a, c) =P (b | c), for all values a, b, c

* Intuitive interpretation:

P (a | b, c) = P (a | c) tells us that learning about b, given that we already know c, provides no change in our probability for a, i.e., b contains no information about a beyond what c provides

* Can generalize to more than 2 random variables
* E.g., K different symptom variables X1, X2, … XK, and C = disease

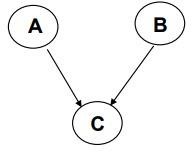
– P (X1, X2…. XK | C) = Π P (Xi | C)

* Also known as the naïve Bayes assumption

**Example of a simple Bayesian network**

p (A, B, C) = p (C|A, B) p(A)p(B)

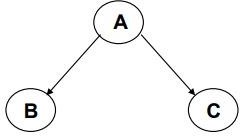
* Probability model has simple factored form
* Directed edges => direct dependence
* Absence of an edge => conditional independence
* Also known as belief networks, graphical models, causal networks
* Other formulations, e.g., undirected graphical models



**Examples of 3-way Bayesian Networks**

Marginal Independence: p (A, B, C) = p(A) p(B) p(C)

Conditionally independent effects: p (A, B, C) = p(B|A) p(C|A)p(A)



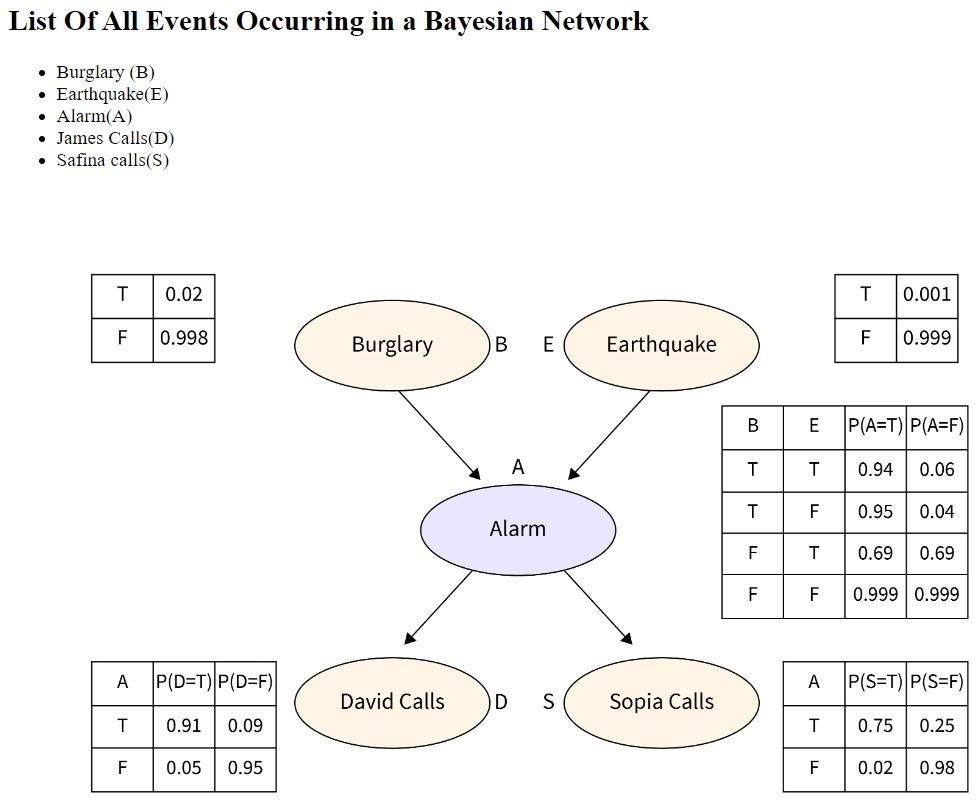
Example 1: Weather Forecast

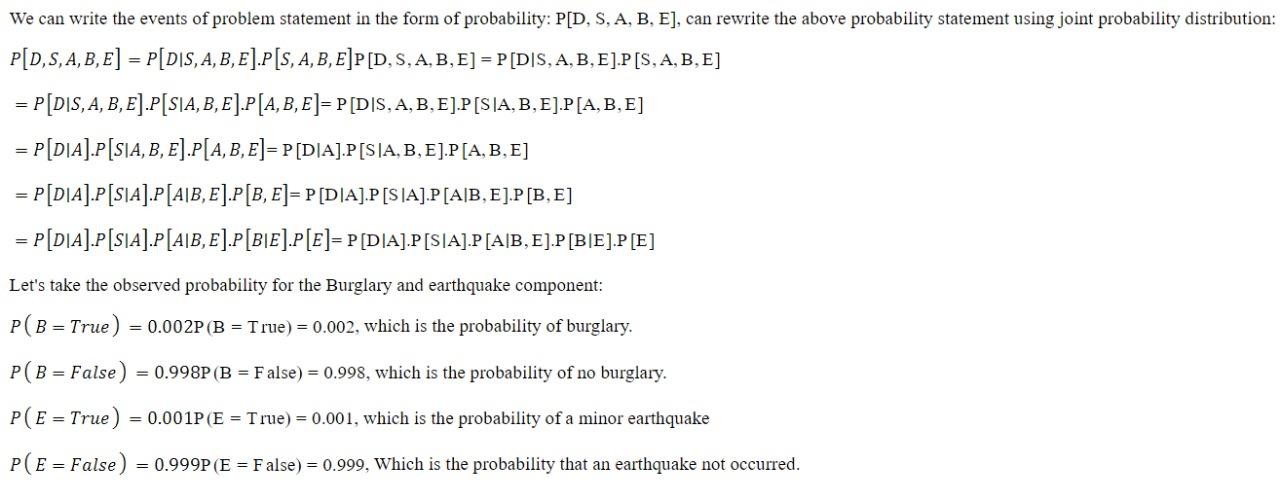
Variables: W (Weather), T (Temperature), P (Probability of Rain).

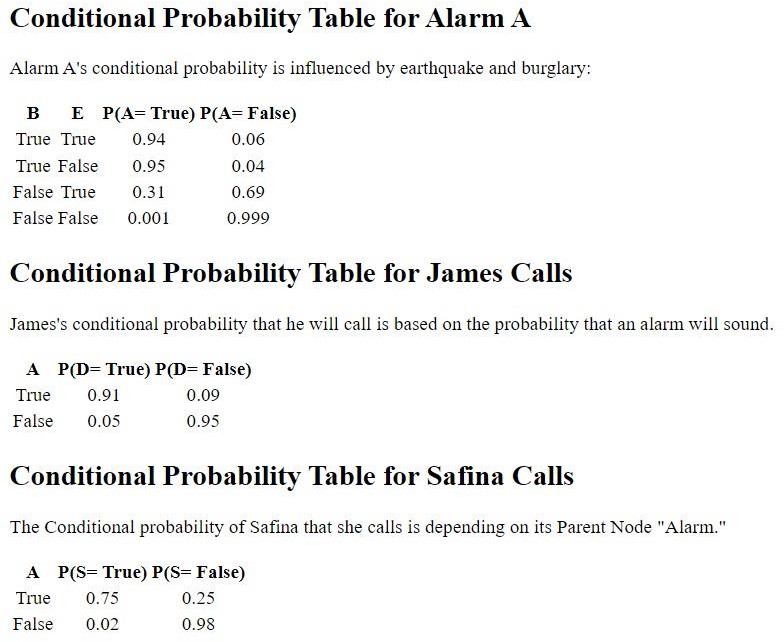
* Dependencies:

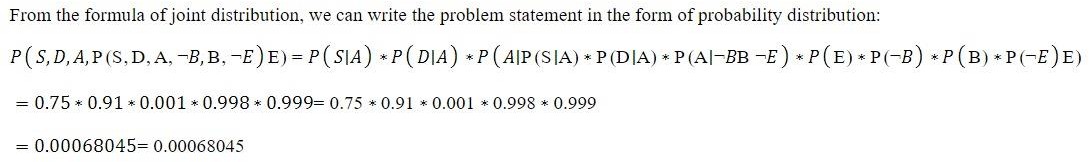
W influences P and T.

T influences P.



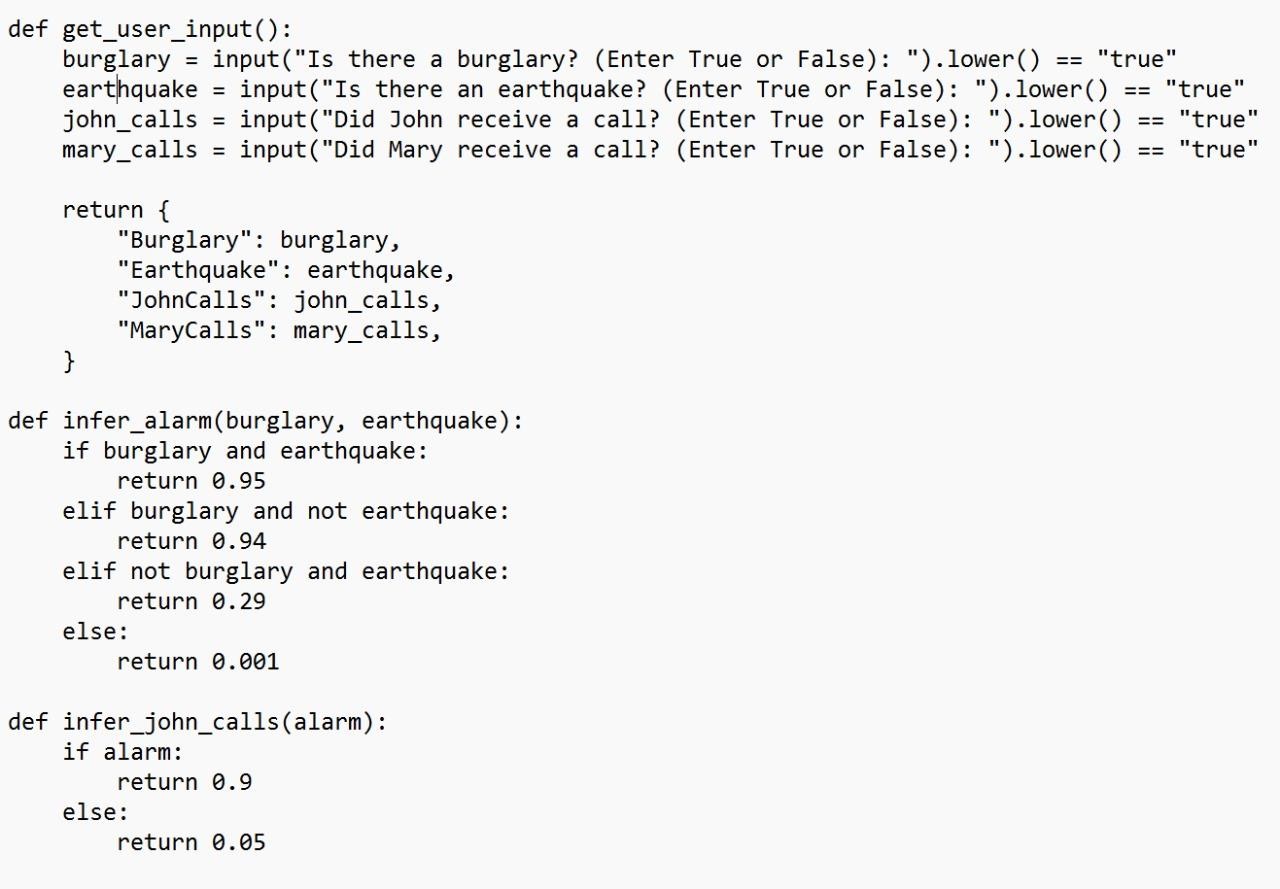


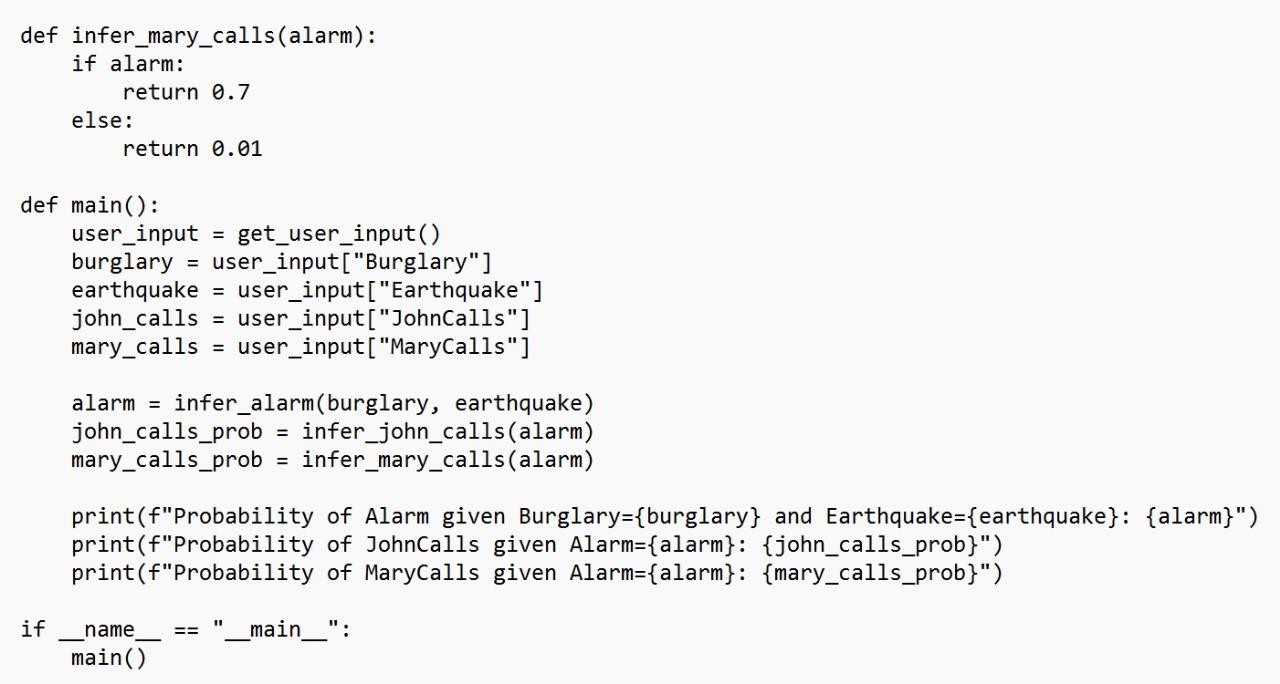




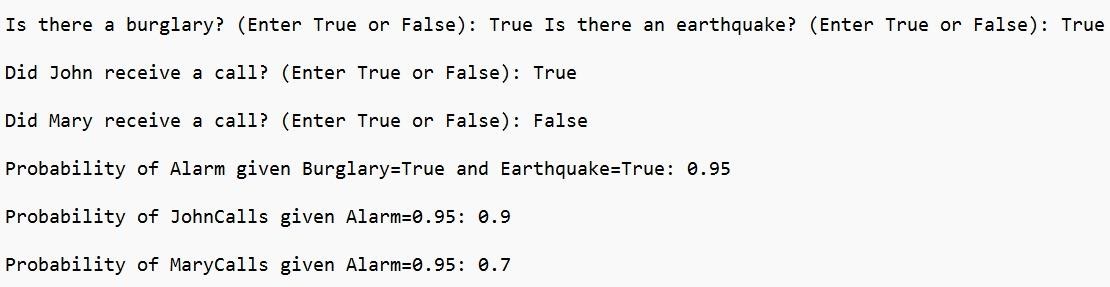
### BAYESIAN RISK ASSESSMENT BURGLARY ALARM

**PYTHON CODE:**





# OUTPUT:



**Applications of BBNs:**

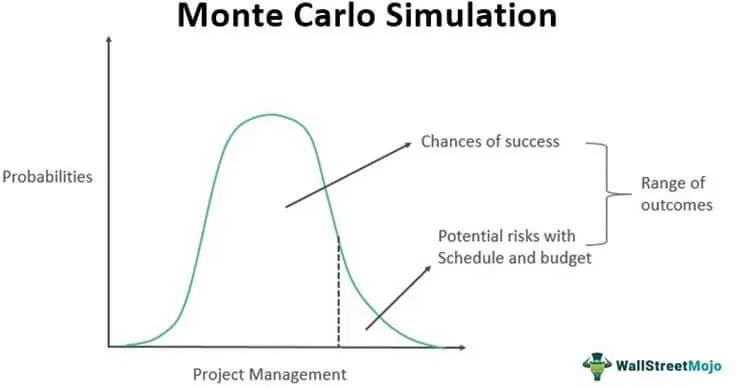
Here are some real-world applications of Bayesian belief networks (BBNs):

* Medical diagnosis: BBNs can be used to calculate the probability of a disease given certain symptoms. This can help doctors make more accurate diagnoses and provide better treatment. For example, a BBN could be used to calculate the probability of a patient having a heart attack given their symptoms, such as chest pain and shortness of breath.
* Spam filtering: BBNs are used in spam filters to identify spam emails. Spam filters work by calculating the probability that an email is spam given certain features, such as the sender's address, the subject line, and the words in the body of the email.
* Weather forecasting: BBNs are used in weather forecasting to update the probability of different weather conditions based on new data, such as measurements from weather stations and satellite imagery. For example, a BBN could be used to calculate the probability of rain tomorrow given the current temperature, humidity, and wind speed.
* Fraud detection: BBNs are used in fraud detection systems to identify fraudulent activity. Fraud detection systems work by calculating the probability that an activity is fraudulent given certain features, such as the type of transaction, the amount of the transaction, and the location of the transaction.
* Network security: BBNs can be used to identify and assess cybersecurity threats. For example, a BBN could be used to calculate the probability of a network intrusion given certain events, such as a failed login attempt or an unauthorized access to sensitive data.
* Risk management: BBNs can be used to assess and manage risks in a variety of contexts, such as finance, insurance, and engineering. For example, a BBN could be used to calculate the probability of a project failing given certain factors, such as the project's budget, schedule, and team.

These are just a few examples of the many real-world applications of BBNs. BBNs are a powerful tool that can be used to solve a wide variety of problems in a wide variety of fields.

**MONTE CARLO SIMULATION**

A Monte Carlo simulation is used to model the probability of different outcomes in a process that cannot easily be predicted due to the intervention of [random](https://www.investopedia.com/terms/r/random-variable.asp) [variables](https://www.investopedia.com/terms/r/random-variable.asp). It is a technique used to understand the impact of risk and uncertainty.



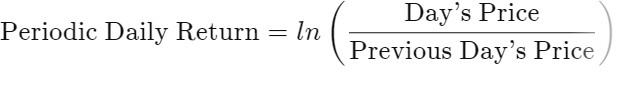
The Monte Carlo simulation was named after the gambling destination in Monaco because chance and random outcomes are central to this modelling technique, as they are to games like roulette, dice, and slot machines.

The technique was initially developed by Stanislaw Ulam, a mathematician who worked on the Manhattan Project, the secret effort to create the first atomic weapon. He shared his idea with John Von Neumann, a colleague at the Manhattan Project, and the two collaborated to refine the Monte Carlo simulation.

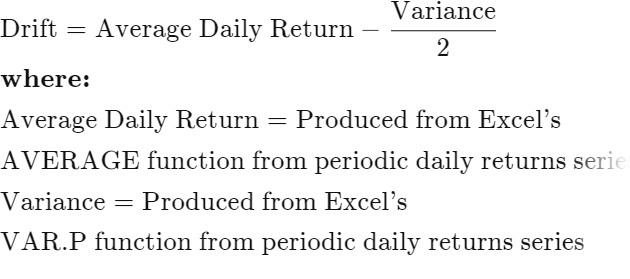
The 4 Steps in a Monte Carlo Simulation:

To create a Monte Carlo simulation, consider the following four steps:

**Step 1:** To project one possible price trajectory, use the historical price data of the asset to generate a series of periodic daily returns using the natural logarithm (note that this equation differs from the usual percentage change formula):



**Step 2:** Next use the AVERAGE, STDEV.P, and VAR.P functions on the entire resulting series to obtain the average daily return, standard deviation, and variance inputs, respectively. The drift is equal to:



Alternatively, drift can be set to 0; this choice reflects a certain theoretical orientation, but the difference will not be huge, at least for shorter time frames.

**Step 3:** Next, obtain a random input:

The equation for the following day's price is:

Next Day’s Price = Today’s Price x e^(Drift+Random+Value)

**Step 4:** To take *e* to a given power *x* in Excel, use the EXP function: EXP(x). Repeat this calculation the desired number of times. (Each repetition represents one day.) The result is a simulation of the asset's future price movement.

By generating an arbitrary number of simulations, you can assess the probability that a security's price will follow a given trajectory.

#### Random Variable:

The random variable is a real-valued function, defined over a sample space associated with the outcome of a conceptual chance experiment. Random variables according to their probability density function.

#### Random Number:

It is a number in a sequence of number, whose probability of occurrence is same that of any other number in that sequence.

#### Pseudo-random Number:

Random numbers are called pseudo-random number where they are generated by some deterministic process, but have already qualified the pre-determined statistical test for randomness.

Applications

#### #1 – Project Management

The Beta function is the most common [**probability distribution**](https://www.wallstreetmojo.com/probability-distribution/) function used in project management. By using this method, businesses can assess risks associated with the schedule and budget.

The PERT function, on the contrary, indicates a triangular distribution of possible outcomes. It helps identify uncertainties associated with a project if the activity duration changes, making the deadline a random variable.

The triangular distribution indicates that the delay in the task start will lead to its early completion, given the deadline is already mentioned.

#### #2 – Finance

Monte Carlo Simulation in finance works on multiple fronts. Of these, the first one is options valuation. It helps analyse potential risks associated with equity options pricing. It simulates

the fluctuation in underlying share values on multiple price paths to determine the option payoff for different price paths. Averaging these payoffs will give the current option price.

The next is the valuation of a portfolio. This method simulates factors affecting the value of multiple portfolios to assess all possible outcomes. Finally, it determines the overall average value of all simulated portfolios and uses it to calculate the most accurate portfolio assessment.

The third one on the list is the sensitivity analysis performed in [**financial modelling**](https://www.wallstreetmojo.com/financial-modeling/). Here, conducting Monte Carlo Simulation in Excel shows a change in a business’ [**net present**](https://www.wallstreetmojo.com/net-present-value-npv-formula/)[**value**](https://www.wallstreetmojo.com/net-present-value-npv-formula/) (NPV) with changes in underlying variables.

#### #3 – Business

In addition to the above domains, the technique assists in project investments and [**default**](https://www.wallstreetmojo.com/default-risk/)[**risk**](https://www.wallstreetmojo.com/default-risk/) assessments of a business. Corporate decision-makers use this strategy to forecast sales volume, commodity prices, labour costs, exchange rates, and risks associated with contract cancellation or tax legislation changes.

#### #4 – Science & Engineering

The Monte Carlo method evaluates the degree of risks and error percentage in various fields, including materials science, engineering, biology, quantum physics, and architecture. The repetitive events and several calculations involved in these processes make the computation complex, but results obtained through this method help arrive close to accurate figures.

## Gabbling using Monte Carlo Simulation

#### Python code:

import random

import matplotlib.pyplot as plt

#1.create a rollDice Function to determine whether the user wins or not #2.create a simple bettor to and import the random winners and losers into it #3.checking results

#4.create a doubler better where everytime we lose we double our betting amount #5.comparing and analyzing both simpler bettor and doubler bettor

#6.Both Statergies lose the same amount but gains are different

#7.for a short time frame life expectency simple bettor is better but for a long-time frame doubler bettor is better

#8.calculating the bustand profits for the bettors #9.using Multiple bettor for using monte Carlo simulator lower\_bust = 31.235

higher\_profit = 63.208

samplesize = 1000

startingfunds = 10000

wagersize = 100

wagercount = 100

da\_busts = 0.0

da\_profits = 0.0

multiple\_busts = 0

multiple\_profits = 0

simple\_busts = 0

simple\_profits = 0

doubler\_busts = 0

doubler\_profits = 0

#the simple bettor will win in short term but in a long term they will lose def rollDice():

roll = random.randint(1, 100) if roll == 100:

return False elif roll <= 50: return False

elif 100 > roll > 50:

return True #50/50 odds Statergy

#every time you lose or win we are increasing the wager size def dAlembert(funds, initial\_wager, wager\_count):

global da\_busts global da\_profits value = funds

wager = initial\_wager current\_wager = 1 previous\_wager = 'win'

previous\_wager\_amount = initial\_wager while current\_wager <= wager\_count:

if previous\_wager == 'win': if wager==initial\_wager:

pass else:

wager-=initial\_wager

print('Current Wager:',wager,'Value:',value) if rollDice():

value+=wager

print('we won,current value:',value) previous\_wager\_amount=wager

else:

value-=wager previous\_wager='loss'

print('we lost,current value',value) previous\_wager\_amount=wager

if value <=0: da\_busts+=1 break

elif previous\_wager == 'loss': wager=previous\_wager\_amount + initial\_wager if(value-wager)<=0:

wager=value

print('lost the last wager,current wager:',wager,'value',value) if rollDice():

value+=wager

print('we won current value:',value) previous\_wager\_amount=wager previous\_wager='Win'

else:

value-=wager

print('we won current value:',value) previous\_Wager\_amount=wager

if value<=0: da\_busts+=1 break

current\_wager+=1

if value > funds: da\_profits+=1

dAlembert(startingfunds,wagersize,wagercount)

def multiple\_bettor(funds, initial\_wager, wager\_count): global multiple\_busts

global multiple\_profits value=funds wager=initial\_wager wX=[]

vY=[]

current\_wager=1 previous\_wager='Win'

previous\_wager\_amount = initial\_wager while current\_wager <= wager\_count:

if previous\_wager == 'Win': if rollDice():

value += wager wX.append(current\_wager) vY.append(value)

else:

value -= wager previous\_wager = 'Loss'

previous\_wager\_amount = wager wX.append(current\_wager) vY.append(value)

if value <= 0: multiple\_busts += 1 break

elif previous\_wager == 'Loss':

#print 'we lost the last one so we will be smart and double' if rollDice():

wager = previous\_wager\_amount \* random\_multiple if(value-wager)<0:

wager=value

#wagering only what we left not going into the negative value += wager

wager = initial\_wager previous\_wager = 'Win' wX.append(current\_wager) vY.append(value)

else:

wager = previous\_wager\_amount \* random\_multiple if(value-wager)<0:

wager=value value -= wager

previous\_wager\_amount = wager wX.append(current\_wager) vY.append(value)

if value <= 0: multiple\_busts += 1

break previous\_wager = 'Loss'

current\_wager += 1

plt.plot(wX, vY,color)# cyan color for doubler bettor if value > funds:

multiple\_profits+=1

def simple\_better(funds, initial\_wager, wager\_count,color): global simple\_busts

global simple\_profits value = funds

wager = initial\_wager wX = [] # wager values vY = [] # funds current\_wager = 1

while current\_wager <= wager\_count: if rollDice():

value += wager wX.append(current\_wager) # wager Count vY.append(value) # appending the funds

else:

value -= wager wX.append(current\_wager)

vY.append(value)

current\_wager += 1 if value <= 0:

value = 'Broke' simple\_busts += 1

plt.plot(wX, vY,color) # plotting the graph if value != 'Broke' and value > funds:

simple\_profits += 1

def doubler\_better(funds, initial\_wager, wager\_count,color): value = funds

wager = initial\_wager global doubler\_busts global doubler\_profits wX = [] # wager values vY = [] # funds current\_wager = 1 previous\_wager = 'Win'

previous\_wager\_amount = initial\_wager while current\_wager <= wager\_count:

if previous\_wager == 'Win': if rollDice():

value += wager wX.append(current\_wager)

vY.append(value) else:

value -= wager previous\_wager = 'Loss'

previous\_wager\_amount = wager wX.append(current\_wager) vY.append(value)

if value <= 0: doubler\_busts += 1 break

elif previous\_wager == 'Loss':

#print 'we lost the last one so we will be smart and double' if rollDice():

wager = previous\_wager\_amount \* 2 if(value-wager)<0:

wager=value

#wagering only what we left not going into the negative value += wager

wager = initial\_wager previous\_wager = 'Win' wX.append(current\_wager) vY.append(value)

else:

wager = previous\_wager\_amount \* 2 if(value-wager)<0:

wager=value value -= wager

previous\_wager\_amount = wager wX.append(current\_wager) vY.append(value)

if value <= 0: doubler\_busts += 1 break

previous\_wager = 'Loss'

current\_wager += 1

plt.plot(wX, vY,color)# cyan color for doubler bettor if value > funds:

doubler\_profits+=1

x = 0

while True: multiple\_busts = 0.0

multiple\_profits = 0.0

multiplesampsize = 100000 currentSample=1 random\_multiple=random.uniform(0.1,10.0) while currentSample <= multiplesampsize:

#multiple\_bettor(startingfunds, wagersize, wagercount)

#simple\_better(startingfunds, wagersize, wagercount,'c') #doubler\_better(startingfunds, wagersize, wagercount,'k') currentSample+=1

if ((multiple\_busts/multiplesampsize)\*100.00 < lower\_bust) and ((multiple\_profits/multiplesampsize)\*100.00 > higher\_profit):

print('#####################')

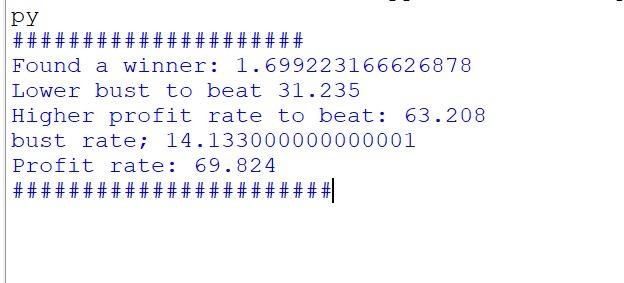
print('Found a winner:',random\_multiple) print('Lower bust to beat',lower\_bust) print('Higher profit rate to beat:',higher\_profit)

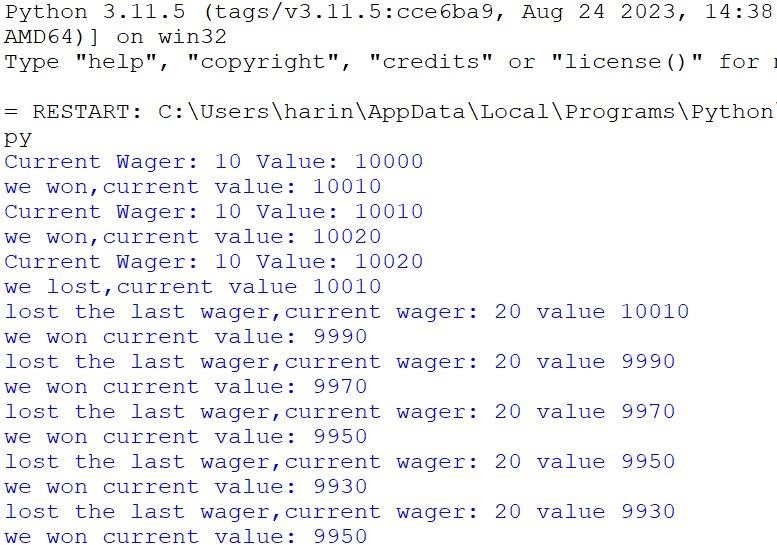
print('bust rate;',(multiple\_busts/multiplesampsize)\*100.00) print('Profit rate:',(multiple\_profits/multiplesampsize)\*100.00) print('#######################') '''print('#####################')

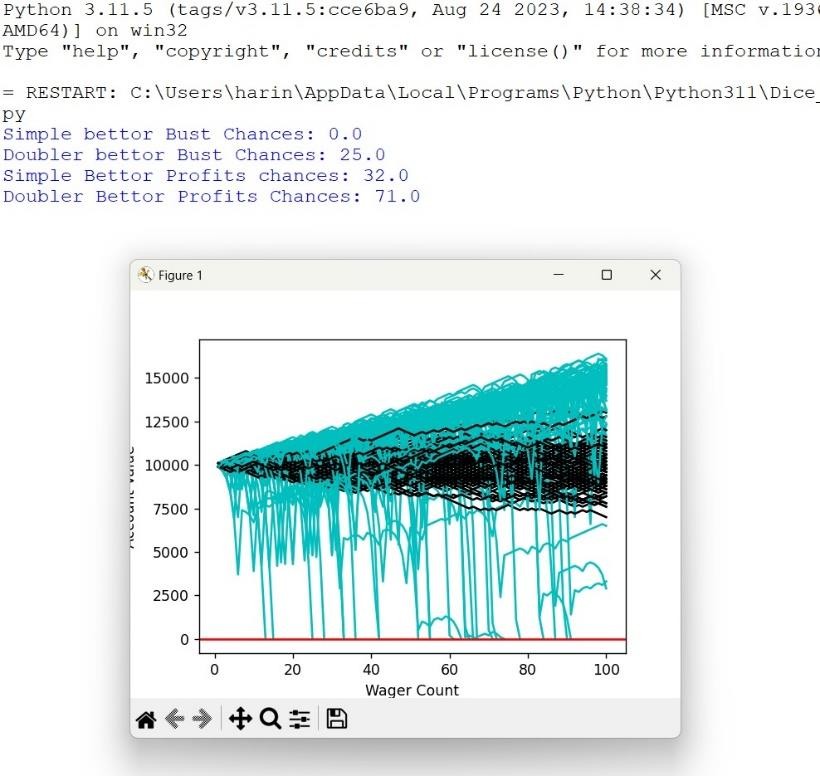
print('Found a loser:',random\_multiple) print('Lower bust to beat',lower\_bust) print('Higher profit rate to beat:',higher\_profit)

print('bust rate;',(multiple\_busts/multiplesampsize)\*100.00) print('Profit rate:',(multiple\_profits/multiplesampsize)\*100.00) print('#######################')'''

#### OUTPUT:







The code is a simulation of different betting strategies in a gambling scenario. Here's a brief explanation of the key components and concepts:

Variables and Constants:

lower\_bust: The lower threshold for bust percentage. higher\_profit: The higher threshold for profit percentage. samplesize: Number of simulations to run.

startingfunds: Initial funds for each bettor. wagersize: Initial wager size for each bettor.

wagercount: Number of wagers placed by each bettor. Various counters for busts and profits for different strategies. rollDice Function:

Simulates a dice roll and returns True for a win and False for a loss based on a 50/50 chance. dAlembert Function:

Implements the D'Alembert betting strategy. Increases wager after a loss and decreases after a win.

Tracks busts and profits. multiple\_bettor Function:

Simulates a betting strategy with variable wager sizes.

Doubles the wager after a loss. Tracks busts and profits. simple\_better Function:

**Simulates a simple betting strategy.** Constant wager size, no adjustments. Tracks busts and profits. doubler\_better Function:

Implements a strategy where the wager is doubled after a loss. Tracks busts and profits.

Simulation Loop:

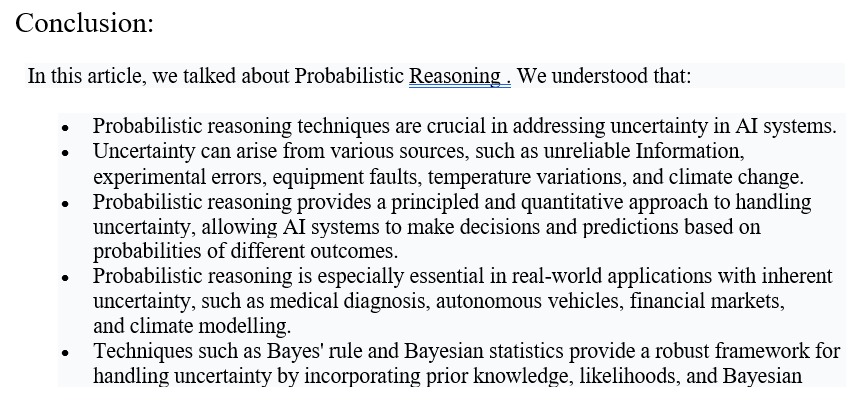
Uses a while True loop to continuously run simulations. Runs simulations for each strategy and compares the results. Prints information when a successful strategy is found.

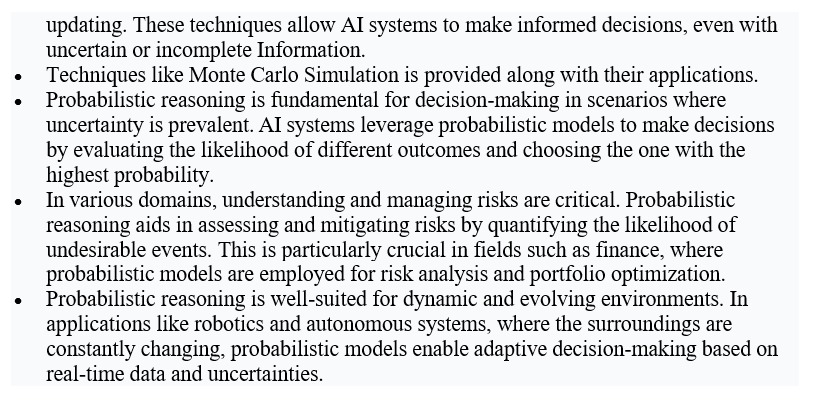
Graphical Representation:

Uses matplotlib to plot graphs showing the performance of different strategies.

Output:

The code continuously runs simulations and prints information when it finds a strategy that meets certain criteria for busts and profits.





References:

[1] Yasmin Yousseff: Bayes theorem and real life applications.

[2] Girija Nandan Kar: Probability and its use in artificial intelligence

[3] Shafi: The power of probability in AI.

[4] Javad Ibrahimli: Use of Bayes’ theorem in autonomous vehicles.

# [5] Adam Hayes, Reviewed by Margret James, Facts checked by Yarilet Perez: Probability Distribution Explained: Types and Uses in Investing.

# [6] [Stefan Hoermann](https://ieeexplore.ieee.org/author/38234688800); [Daniel Stumper](https://ieeexplore.ieee.org/author/37086063542); [Klaus Dietmayer:](https://ieeexplore.ieee.org/author/37283417900) Probabilistic long-term prediction for autonomous vehicles.

# [7] Adekitan Aderibigbe: Monte Carlo Simulation.

# [8] Pedro Domingos: A few useful things to know about Machine Learning.

# [9] Julien Bert, David Sarrut: Monte Carlo Simulation for medical and biomedical applications.

# [10] Peter Barbrook-Johnson, Alexandra Siobhan Penn: Bayesian Beliefs Network.