

King's College London, Strand Campus, London, United Kingdom

**Faculty of Natural and Mathematical Sciences**

# Department of Engineering/Informatics

7CCSMPRJ/7CCSMUIP (Individual Project 21~22 NSY 000001)

Individual Project Submission 2022/08/08 (Replace by 2021)

Name: JINYU MIAO

Student Number: 21030339

Degree Programme: Artificial Intelligence MSC

Project Title:

Supervisor: Elizabeth Black

Word count: XXXXXXXX

RELEASE OF PROJECT

Following the submission of your project, the Department would like to make it publicly available via the library electronic resources. You will retain copyright of the project.

RI **agree** to the release of my project

I **do not** agree to the release of my project

**Signature:** (An e-signature is acceptable, i.e., attach your signature as a picture, sign using the draw function)

Date: 08/08/2022

Department of Informatics

King’s College London

United Kingdom

7CCSMPRJ/7CCSMUIP MSc Project

~~A Rail Network Passenger Flow Distribution System~~

JINYU MIAO

Student Number: 21030339

Degree Programme: Artificial Intelligence MSC

Supervisor’s Name: Elizabeth Black

**This dissertation is submitted for the degree of MSc in ~~A RAIL NETWORK PASSENGER FLOW DISTRIBUTION SYSTEM~~**

**Originality Avowals**

I verify that I am the sole author of this report, except where explicitly stated to the contrary. I grant the right to King’s College London to make paper and electronic copies of the submitted work for purposes of marking, plagiarism detection and archival, and to upload a copy of the work to Turnitin or another trusted plagiarism detection service. I confirm this report does not exceed 15,000 words.

Acknowledgement

First, I would like to express my sincere gratitude to my supervisor for her professional instruction and encouragement through each stage of the process. Without her understanding, patience, and helpful suggestions, I might not complete my dissertation successfully. She was very patient in every meeting and gave pertinent suggestions, making me much more confident and not afraid of making mistakes. Sincere gratitude should also be extended to other faculty members who helped me with my dissertation, such as my personal tutor, professors, and teaching assistant.

Equally important, I would particularly like to be grateful to my school, including my GP, my psychological counselor, NHS staff, and the information department staff. I went through a challenging period and thought of committing suicide several times. Their sincerity and patience made me feel warm and hopeful and firmly believe that there was a strong school team behind me that I could be proud of. In the future, I will also be sincere to everyone in society.

Last but not least, I would like to thank my parents, friends, especially my partner, and roommates for their love and support, who helped me through a hard time and gave me confidence during the whole postgraduate journey.

Abstract

It is a precis of the report (normally in one page), which should include:

* A brief introduction to the project objectives
* A brief description of the main work of the project
* A brief description of the contributions, major findings, results achieved and principal conclusion of the project

Contents

(It will be generated automatically finally)

List of Tables

List of Figures

1. Introduction

In recent years, the development of artificial intelligence has covered many fields. As one of the most attractive research interests, argumentation models and techniques has been widely paid attention by researchers. These applications include, for example, multi-agent systems and artificial intelligence for legal reasoning. The fields of qualitative reasoning technology, rough set theory, evidence theory, and others have made recent advances in artificial intelligence. A new generation of intelligent decision-making methods have been developed by applying these theories and methods to the decision-making process. A decision support system (DSS) is a tool for analyzing and solving semi-structured and unstructured problems. Intelligent decision methods enhance this ability as well as impact the architecture and notion of decision support. The successful combination of artificial intelligence and decision support system in various parts of intelligent decision support system has been showed by some research papers [1-2]. The examples include systems for managing data, modeling, methods, knowledge bases, and management systems, which are definitely helpful in improving decision support.

Argumentative decision-making method provides a common sense model for the kind of judgment, which can be customized for specific databases and error types. It uses a knowledge-based approach to automate the process of data cleaning: the user implicitly includes the background knowledge of the database and any problems that may arise when defining the data.

For example, in one scenario, what if someone says they live in Beverly Hills without leaving any other information? We should pick up the names of the apartment all around the country in the database. Although there is the famous Beverly Hills in California, there is also one in Florida, Missouri and Texas. We should also know that a community called Beverly Hills is located in in Baltimore. How do we know where this man lives? This is where argumentative decision-making comes into play.

In this system, users can provide argumentative-decision-making system with background knowledge about the domain and how the data might be corrupted. This system combines the knowledge with commonsense probabilistic reasoning to present the answer. For example, if more information is provided, such as the distribution map of the rent prices per month for the apartment in the country, we will know which one is the answer to the previous question. For example, the model could infer that the correct answer is Beverly Hills in California because of the high cost of rent where the respondent lives.

In addition to the points we discussed above, we should consider the argumentation in decision support systems as well. Conflicts arising from beliefs, assumptions, viewpoints, opinions, and a wide range of other mental attitudes are often resolved by mental arguments. The use of arguments in favor and against a given position is often our go-to response when we are faced with an incomplete or inconsistent piece of information. Cooperation or competition during an argument is not uncommon, aiming to reach a conclusion or defend or advance one's own position.

Developing artificial argumentation systems is as diverse and exciting as creating artificial intelligence itself, since argumentation permeates both inside and outside of the mind. We can use models, tools and autonomous artificial agents to carry out cognitive tasks in response to this valuable phenomenon. In the near future, computational models of argument are likely to become a highly interdisciplinary research area due to the investigation of a number of interesting lines of research in artificial intelligence and neighboring fields. In this area, numerous aspects of human intelligence are expected to be better understood and modelled.

Researchers are developing a computational model that explains how human beings create, exchange, analyze, and use arguments in the presence of uncertain, incomplete, or inconsistent information [3-4]. It is impossible to reduce the vast body of literature to a single reference framework because arguments manifest in a variety of ways in real life and relevant models do too. Even so, identifying some basic components for constructing a model of argumentation can be considered to be the basis for constructing a model. Different modeling approaches use different layers, combine different layers in different ways, and formalize different layers in different ways.

According to research conducted by Anaconda **[x]**, cleaning data can take up to a quarter of a data scientist's time. How to automate this task has always been a challenging task. This is because different datasets require different types and levels of cleanup and the cleanup process often relies on common sense to determine what objects in the world are.

80% of the work in AI models is on the data, and data cleaning is a key step to ensure the quality of the model. It is often difficult to automate so we build a Argumentative-decision-making project to manually cleaning data.

In this project, we will do some research by using the text files we selected. We will use Python as the programming language and the Jupyter Notebook is the environment. The main contribution of this project will be listed as following:

1. Creating an argumentation-based decision support system to aid decision makers in dealing with the contentious nature of inconsistent knowledge bases
2. Doing some experiments which evaluate the performance and prove the effectiveness of the system
3. Applying reinforcement learning algorithms in this system and make comparison.
4. We use the automatic cleaning tools with Bayesian models.
5. Background

Argumentative decision making system is developed by the researchers in the probability of Computing Project (Probabilistic Computing Project) for specific areas of the probability of a programming language. It is designed to simplify the development of the application of artificial intelligence and automation.

* 1. Decision support system

 Business or organizational decision-making is supported by a decision support system (DSS). An organization's DSS serves the management, operation, and planning levels (usually mid- and upper-level management) to help people make decisions about rapidly changing problems and problems that cannot be exactly predicted in advance, such as those that are unstructured or semi-structured. For example, the photos, videos, and audio files are unstructured data and the email data are semi-structured. Either computerized or manually-driven decision support systems can be used. A DSS is a system that works with knowledge that requires integration of raw data, documents, and personal knowledge in an interactive software application in order to identify, solve problems and make decisions more efficiently.

AI technology has become increasingly integrated into decision support system architectures and problem solving methods, affecting decision-making methods and processes directly in recent years. Over the years, the research on intelligent decision support systems has gradually developed with various applications including agriculture [5], supply chain [6], and finance [7].

In a wide range of industries, decision support systems are used. Some examples are:

Routing using GPS [8]. The shortest and best route can be planned by analyzing the options available. Real-time traffic monitoring is often included in these systems in order to avoid congestion.

Planned-growth systems [9]. In order to plant, fertilize, and reap their crops at the best time, DSS provides farming information. In addition to creating "virtual factories" for its corn manufacturing operations, Bayer Crop Science is applying analytics and decision-support across every aspect of its business.

The clinical decision support system (DSS) [10]. ICU patients at Penn Medicine can now be taken off their ventilators faster by using a clinical DSS.

Dashboards for ERP [11]. Managers can view performance indicators from dashboards. To identify which agents need additional assistance, Clearlink, a marketing company specializing in digital marketing, uses a DSS system.

The three key components of decision support systems include a database, software system, and user interface. Multi-sourced data is collected, including data gathered from inside and outside the organization, and data generated by applications. Databases used for DSS may vary in size depending on how large the data warehouse is or how small a standalone DSS is. A digital decision-making model (including user-specific criteria and decision context) is used to build the software. The number and type of models used in the DSS depend on the system's purpose. Models such as the User Interface are common. Users can interact with the DSS using displays and other interfaces. Users can interact with and view results through dashboards and other user interfaces.

* 1. Arguments and argumentation

Philosophical argument is at the center of the subject. The practice of refuting claims with arguments has motivated philosophers for millennia, as have discussions of the nature of arguments and arguments. Moreover, argumentative practices are commonly used in diverse settings, including scientific inquiry, court proceedings, educational institutions, and government institutions. Philosophical, linguistic, legal, cognitive and computer sciences, political science, and other disciplines are incorporated into the study of argumentation.

Identification, analysis, evaluation, and invention are the four elements of argumentation [12]. In the argumentation process, one can find the premises and conclusions of an argument. When attempting to determine the strength of an argument, one must describe the structure of arguments. Native-language argumentation has the tendency to implicitly state at least an antecedent and a conclusion. Traditional definitions of such arguments refer to them as enthymemes. Evaluation involves assessing an argument's strength and weakness based on general criteria. Inventing new arguments is the process of proving a certain point using new evidence. Nonetheless, some work has been done on the fourth task, largely with Aristotelian topics, although the first three tasks have received much attention.

* 1. Applications

1. In recent years, there are many applications for decision making systems. The first area is corporate management. Corporate functional management consists of applications, which all the corporate functional management applications are dominated by applications of production and operations management (POM). The example consists of marketing information system, financial applications, and strategic management. Other application fields include education, government, agriculture, urban planning, and medical usage.
2. Literature review

DSS is short for Decision Support System. The concept of DSS was first proposed by Scott Morton and Keen in the mid-1970s [5].  It is a new management system that emerged in the late 1970s. The core of management is decision making. Decision making is a process. DSS is a theory and method of applying decision science and related subjects with computer as tool. The information system which assists decision makers to solve semi-structured or unstructured decision problems by means of human-computer interaction is a scientific tool to assist decision making in a specific form. It provides decision-makers with knowledge and initiative through man-machine dialogue. A combination of creativity and information processing skills. Qualitative and quantitative work environment to assist decision makers to analyze problems and explore decision-making methods. Evaluate, forecast and select.

After more than 20 years of development, DSS has made remarkable achievements in theoretical research and system development, showing a trend of diversified development [6].

In the process of DSS development, DSS gradually expands towards unstructured problem domain [7], meanwhile, artificial intelligence means and technologies are constantly introduced, and knowledge components are appropriately added to realize the perfect combination of DSS and expert system. The idea of the combination of decision support system and expert system was put forward in the early 1980s. Because of this, the initial model of intelligent decision support system was constructed. Intelligent decision support system belongs to the integration of numerical analysis and knowledge processing, so it has the advantages of quantitative analysis technology of traditional decision support system and symbol processing of expert system. Compared with decision support system, it can better deal with semi-structured and unstructured related problems through proper integration [8]. In the research of intelligent decision support system, the focus is on the traditional decision support system and expert system, such as the concrete structure, integration mode and interface technology of the system. The birth of intelligent machine and neural network enables new technologies such as database knowledge discovery to be widely applied in decision support system, thus greatly improving the support efficiency of intelligent decision support system [9]

With the rapid development of decision support system, people constantly put forward new requirements, and pay more attention to get computer support in high decision level and complex decision environment. The main object of computer support is not only a single decision-maker or a decision-making group, but also independent and closely related decision-making organizations. Many large scale management decision-making activities cannot use centralized manner, so that these activities involve different liability policy makers, in the process of actual decisions need information resources fully, at the same time make many important decision factors to a greater range of motion, this is the kind of organization or distribution decisions [10]. Distributed decision support system (DSS) is an extension of traditional centralized decision support system (DSS) as well as a combination of distributed decision, distributed system and distributed support. Combined with its specific concept analysis, distributed decision support system is mainly a computer network composed of several physically separated information processing nodes, and at the same time, each node in the network has a decision support system or has multiple decision-making assistance functions. The main advantages of this distributed decision support system are reliability, high efficiency, easy operation and resource sharing [11].

**Recent years have seen the application of decision support systems in the transportation industry...**

1. Fundamental Concepts
   1. PGM programming

The first section of this chapter is PGM programming. In our project, we chose the placename.csv to experiment with Bayesian cleaning, the results of which has been shown in Figure 4.1. We can explore the data in Figure 4.1 which plots the distribution of text word count of all mentioned place names which are related to the chosen data. Figure 4.2 illustrates the place of frequency distribution of the datasets we chose. This table plots two places with largest frequency: London Central Hostel and Coca-cola London Eye. By plotting the histogram with frequency, we may directly see the number of each variable or place with the frequency.

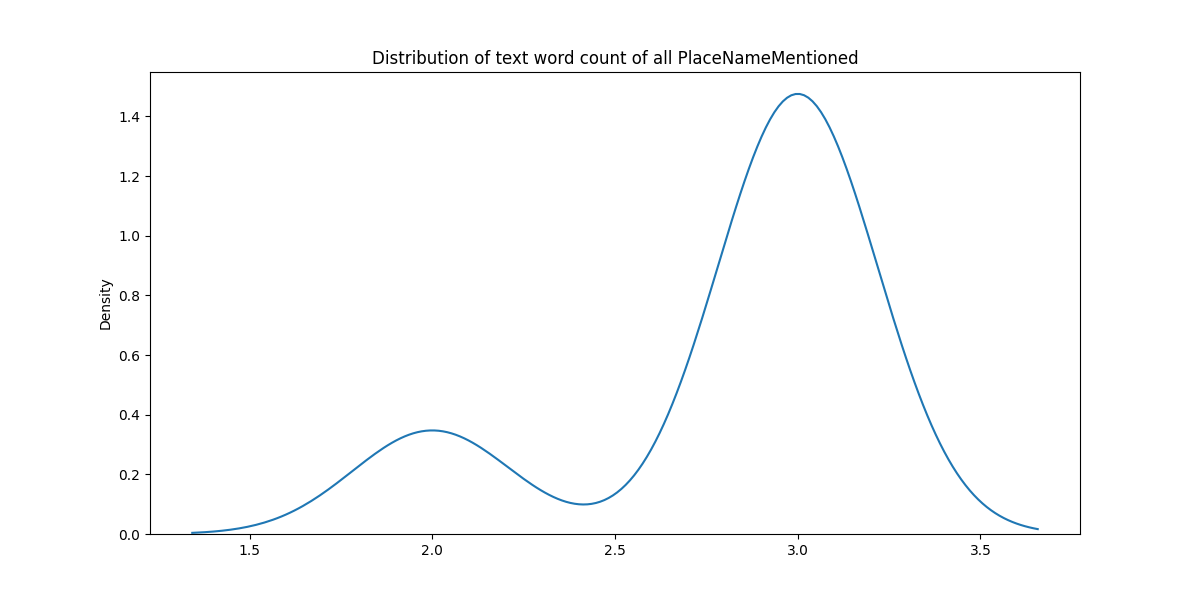


Figure 4.1 The distribution of text word count of all mentioned place name

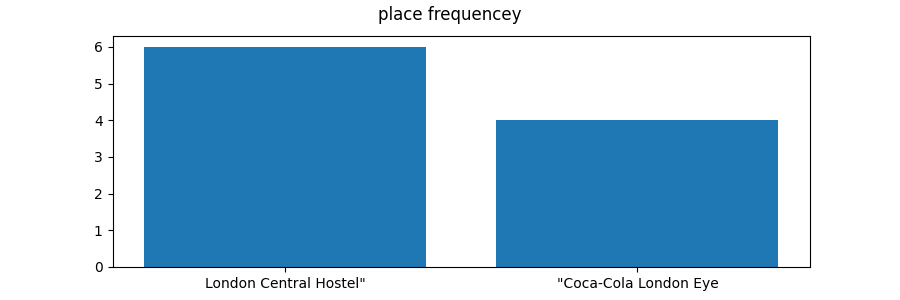
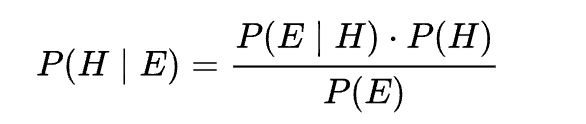


Figure 4.2 Place of frequency distribution

After starting the probabilistic programming, a probabilistic programming language (primarily using Bayesian reasoning) that leverages data-specific knowledge to clean up and normalize dirty data was used in Figure 4.1 and Figure 4.2. In these cases, we can auto-correct and auto-complete these values: NULL values, typos, duplicates, and inconsistencies. The datasets we used is related to the hospitals (clean and dirty) in the and Argumentative-decision-making. A similar Bayesian inference method is used to complete and correct some errors and missing content based on probability graphs. Theoretical support of Bayesian response inference:



where P(A | B) represents the condition assuming that event B is true, event A occurs.<https://zh.wikipedia.org/wiki/%E6%9D%A1%E4%BB%B6%E6%A6%82%E7%8E%87>

Hypothesis whose probability may be influenced by experimental data which can also be referred to as evidence. <https://zh.wikipedia.org/wiki/%E5%AF%A6%E9%A9%97%E6%95%B8%E6%93%9A>Generally, there are many competing hypotheses, and the task is to determine which one is the most likely. Figure 4.3 represents the schematic diagram of Bayesian probabilistic graph inference. In this figure, there are three nodes represented as random variables: A, B, and C. Association between random variables is represented by an edge. The tables below shows the inference results.

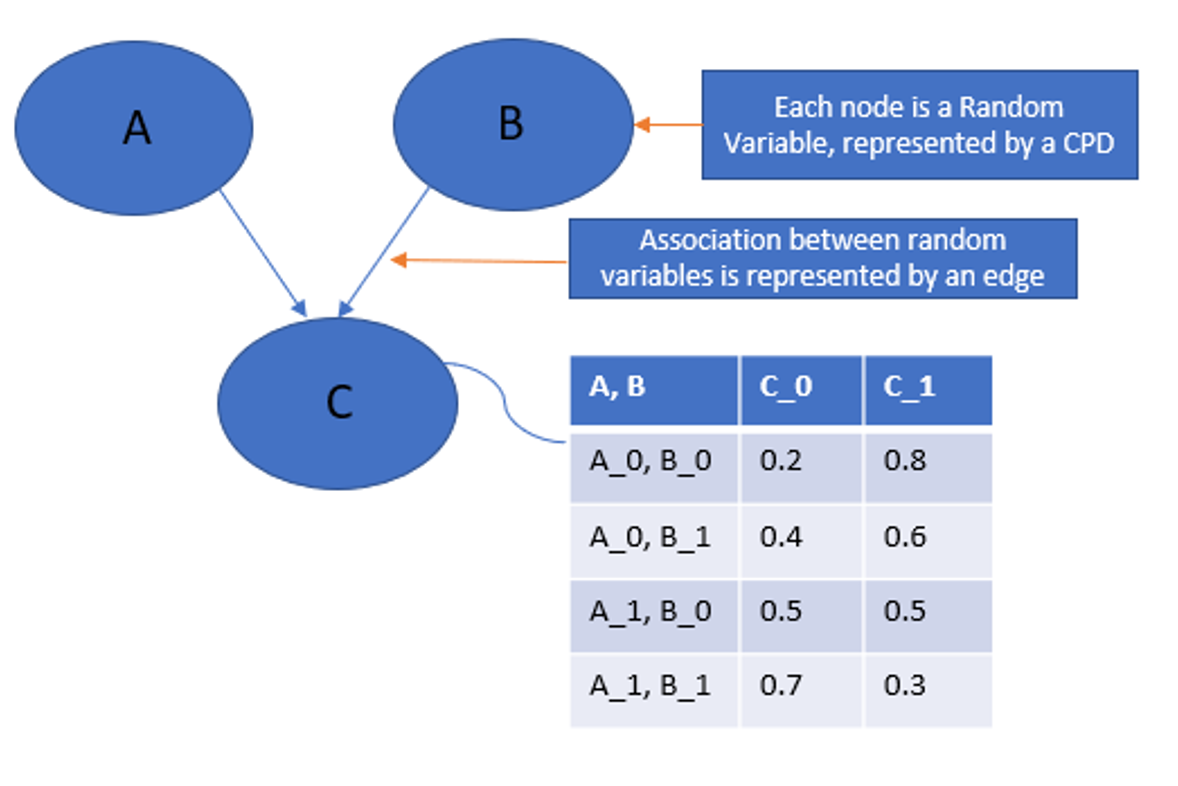


Figure 4.3 Bayesian probabilistic graph inference

Bayesian inference is the key point is to use bayes theorem combined with new evidence and the prior probability of before, to get a new probability (the frequency and the inference on the contrary, frequency theory inference only consider evidence, regardless of the prior probability).

Real-world cases are more complex and more accurate results can be inferred from more information.Actually about evidence search this part, you can also do more, forming a more complicated model, and then finally according to have observed variables to deduce the results we want.

Usually, using probabilistic graphical model based on the graphical representation of as on the basis of the distribution of multi-dimensional coding, and graphics is in the specific distribution of a set of independent compact or decomposition representation.

It is common to use the graphical representation of the two branches of the distribution, that is, both families contain factorization and independence properties, but the set of independence they can encode is different from the factorization of the distribution they induce.

Machine learning a core task of implicit knowledge from the observed data mining, and probability graph model is an approach to achieve the task. PGM cleverly combines graph theory and probability theory. From the perspective of graph theory, the PGM is a figure, contains nodes and edges. Nodes can be divided into two categories: hidden nodes and observation.Edges can be directed or undirected, corresponding to directed and undirected graphs, respectively. We construct a graph using observed nodes to represent observed data, hidden nodes to represent potential knowledge, edges to describe the relationship between knowledge and data, and finally obtain a probability distribution.

After a given probability distribution, there are two tasks:

1. Inference The posterior distribution of hidden nodes is inferred for a given observation node
2. Learn the parameters of this probability distribution to gain knowledge.

<https://zh.wikipedia.org/wiki/%E9%A2%91%E7%8E%87%E5%AD%A6%E6%B4%BE%E6%8E%A8%E6%96%AD>

* 1. The inference of model with prior probability

Our objective is to infer and find some variables from the model of conditional probability distribution is the same, namely, P (Y | E = E), including Y ⊆ χ and E ⊂ χ. Furthermore, if we consider the prediction of the value of a new data point, we are basically trying to find the conditional probability of an unknown variable, given the observed values of other variables. These conditional distributions can be easily calculated from the joint probability distribution of variables by marginalizing and reducing them in terms of variables and states.

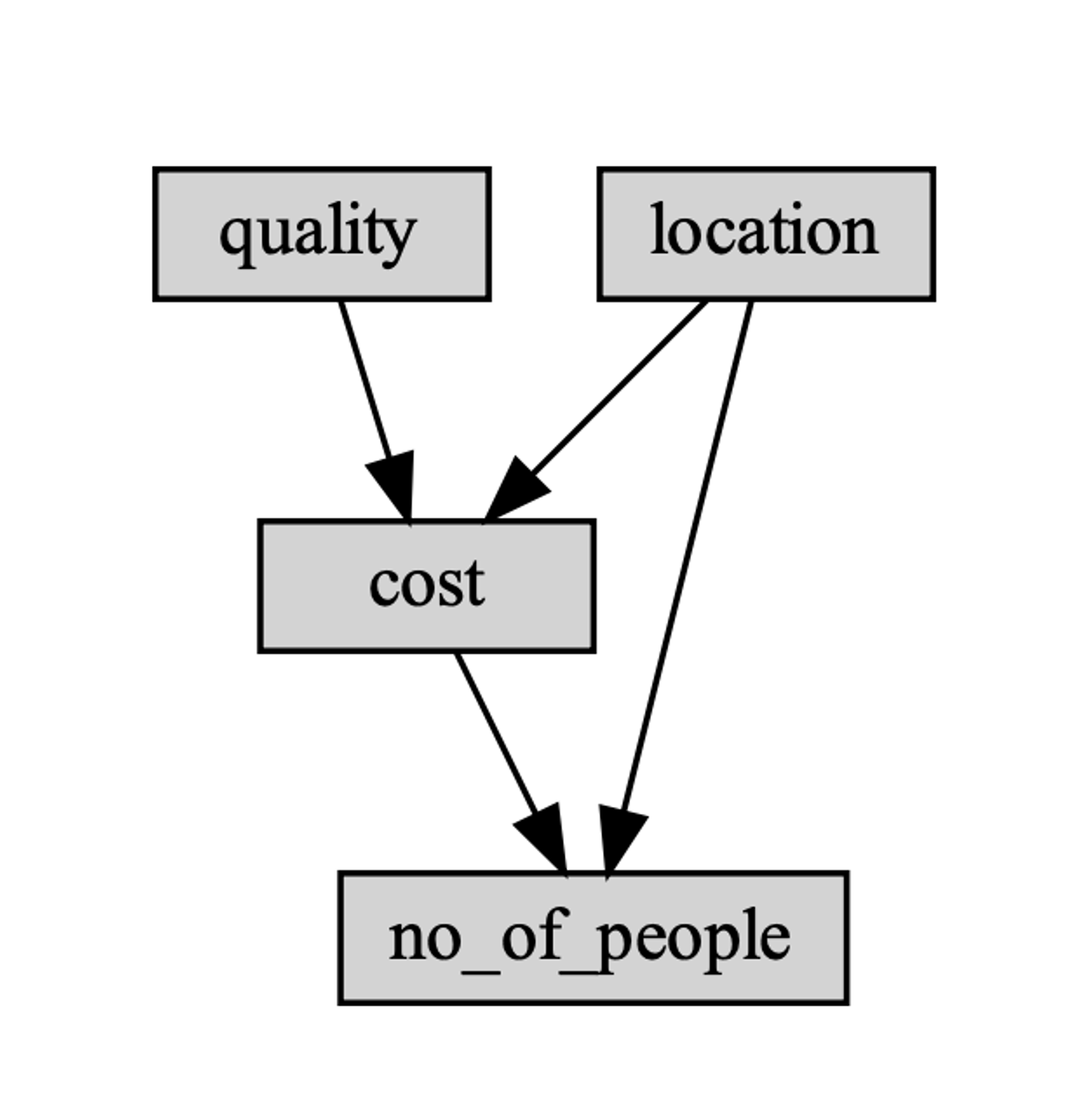
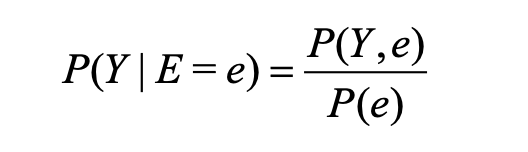
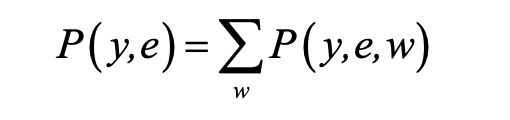


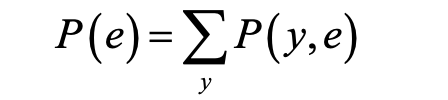
Figure 4.4 The example of hospitals with different locations and quality of service

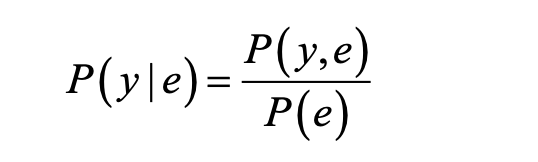
Now we consider the example of hospitals with different locations and quality of service, as shown in Figure 4.4: We consider various inferential queries that can be tried on our model. For example, assuming the position, the high cost, and the quantity, we want to find a restaurant probability of good quality. For example, if patients are likely to come in, which leads to a probabilistic query:

P(Q = good | L= good,C = high,N = high). In addition, if a machine learning problem is considered: the prediction of the number of people to hospital in the case of a given other variables will be on the model of reasoning query and have a higher probability of state will be the model prediction. Now, let's look at how to calculate the conditional probability from the model.









Performing a similar calculation for each state Y of the random variable Y, the conditional distribution on Y can be calculated, given that E = e.

If variables with a joint distribution are given, it can be observed that a conditional distribution can be found. We will discuss when calculating the conditional distribution can help us avoid complete probability distribution of all kinds of algorithms. Now, we will explain what is calculating the complexity of these inferences.

If the worst case is considered, the size of a table in a graphical model, this makes reasoning is a NP hard problem, unfortunately, even the approximate method of calculating conditional distribution is NP - hard. The proof of these results is beyond the scope of this book.

The computational complexity of variable elimination operations is related to the choice of elimination order, and this is related to the elicitation graph **[x]**. Smaller tree widths ensure better complexity than elimination sequences with higher tree widths. Finding the elimination of the minimum tree width is an NP-complete problem, and there is no easy way to find the complexity of network reasoning by simply looking at the network structure. A number of techniques can be used to find a good elimination order: the chord graph property of the induced graph. A graph as a chord graph is defined if it contains no rings of length greater than 3 and if there are no edges between two nonadjacent nodes of each ring. In other words, each minimum cycle in the string graph is three lengths. Every induced diagram is a string diagram. Moreover, the opposite of the theorem holds, that is, every chord plot on these variables corresponds to some elimination order. To find the elimination order, the maximum cardinality search algorithm is used. In this algorithm, we basically iterate χ times, and in each iteration and try to find the variable that has the most labeled variables. Finally, the variable is labeled, which will eliminate sorting.

Figure 4.5 represents the elimination operation process. It shows the information of finding elimination order, eliminating cost, and eliminating quality. Figure 4.6 illustrates the structure of used digraph. It represents the size, node, cost, location, and quality of the graph.

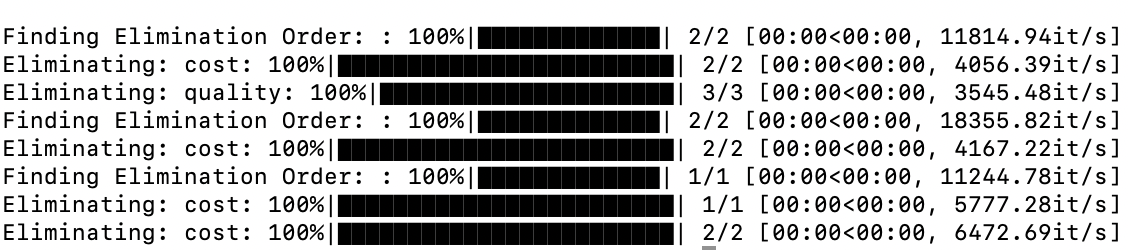


Figure 4.5 Variable elimination operations process

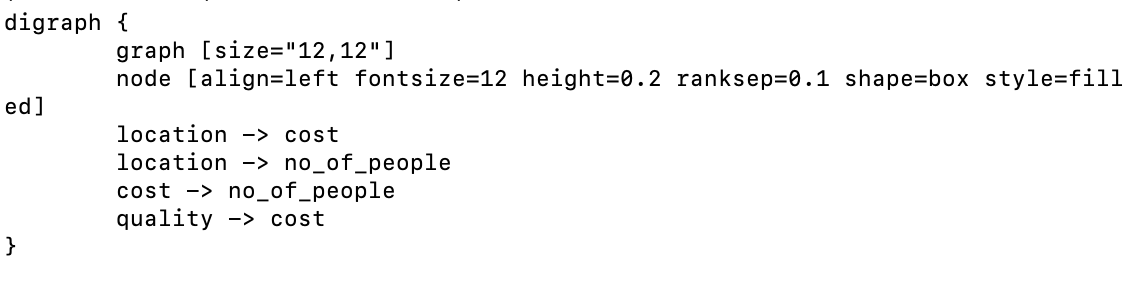


Figure 4.6 The structure of used digraph

* 1. The similarity removal based on machine learning

There are two basic ways to measure the similarity between two sentence vectors: one is to compare the Euclidean distance between them. The smaller the Euclidean distance, the more similar the sentence vectors are. The other is to compare the cosines of the two, and the larger the value, the more similar the two are. The natural metric for KMeans clustering is Euclidean distance minimization, so let's start with the first metric, which is the Sklearn package.

The number of clusters is obtained by combining the intra-cluster similarity, inter-cluster similarity, cliette\_score, calinski\_harabaz\_score, and adjusted\_rand\_score, which sklearn uses to evaluate the cluster results.

We mainly use K-means machine learning method to remove data records that may be duplicated in the data set, so as to reduce the workload of manual inspection and mechanical work of weight removal.

K-means clustering starts by randomly identifying K initial points as centroids (this is also a problem with K-means clustering, where an unreasonable choice of K makes the model ill-suited and poorly interpreted). We then assign each point in the dataset to a cluster. Specifically, we find the nearest center of mass for each point (Euclidean distance, but other distances can also be used) and assign it to the cluster corresponding to the center of mass. Once this is done, the centroid of each cluster is updated to the average of all points in the cluster. This process is repeated until all points in the data set are closest to their center of mass. The specific process has been explained in the following steps:

1. Create k points as starting centroids (randomly selected)
2. When the cluster assignment of any point changes (the algorithm ends if it does not change)
3. Calculate the distance between the centroid and the data point for each data point and center of mass in the dataset.
4. Assign the data point to the nearest cluster.
5. For each cluster, calculate the mean of all points in the cluster and use the mean as the center of mass.

In the training process of similarity comparison, two important links are also very important to the results: 1. whether sentence vectorization is reasonable; 2. whether the measurement standard of sentence similarity is reasonable. For the first point, TfidfVectorizer can transform the original text into the feature matrix of TF-IDF, which lays the foundation for the subsequent text similarity calculation, topic model, text search and ranking, and a series of other applications. Converts a collection of text documents into a sparse matrix of counts. This is implemented internally by calling the scipy.sparse.csr\_matrix module. Also, if CountVectorizer() is called without a prior dictionary and without a profiler that performs some kind of feature selection, the number of feature words will be equal to the vocabulary found by analyzing the data directly with this method. <https://zh.wikipedia.org/wiki/%E5%90%8E%E9%AA%8C%E6%A6%82%E7%8E%87>

1. Implementation of Python code
   1. Argumentation

In this project, there are several Python file for the system. The first one is argumentation.py. It aims to realize argumentation function. Figure 5.1 illustrates the screenshot of the code. The second file is logic.py, which builds a system that assumes logical consistency with prolog syntax. Based on the base logic we used in the project, the system also utilizes the notion of cases and "support.". In cases where the knowledge base contains or implies the literal in the antecedent (body), the rule supports the subsequent literal. This third Python file called knowledgebase.py. It contains the main class that we will import. It contains data structures to read in and represent the contents of simple knowledge bases as collections of Clause and Rule, and Literal instances. In this case, the simple knowledge base is a collection of simple clauses and rules.

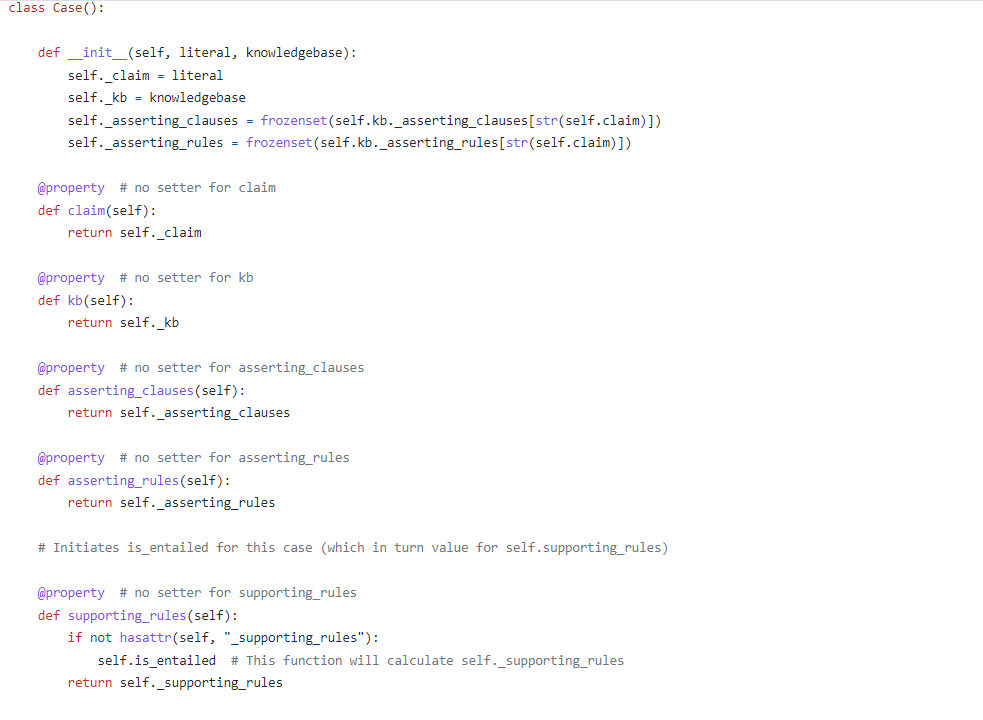


Figure 5.1 The screenshot of the argumentation.py

* 1. Logic

We assume that the decision support system has the following logic components: atoms, literals, negation, clauses, rules, statements, knowledge base (KB), containment, and entailment.

The first logic is atoms. It is a mathematical formula (as in arithmetic), which is also called an atomic formula. A character string that is at least one non-numerical character and an alpha-numerical string (which accepts underscores) identifies an atom. The examples of atoms can be: work, COMP\_dissertation, and decision\_support\_system.

The second one is literals. There are two Booleans: positive and negative for literals. The positive value asserts some atom, which denoted "a" for atom "a". The negative values denoted “~a” for atom “a”, which denoted the logic complement for the atom. As a result, the two possibilities for atoms are: atoms and logic complements. In order to help reader understand the definition, we have shown some examples of literals: a, ~a, John\_passed\_COMP1100, ~John\_passed\_module\_COMP1100, FrankIsHppy, ~FrankIsHappy.

The third basic logic is negation. It can only be used in literal, for example, the “~” in “~Frank\_have\_passed\_the\_module”. It defines the logical compliment.

In addition, we also use causes in the logic.py. Clauses are in the form, where all a\_i are literals the clause asserts in conjunction: a\_1, a\_2, ..., a\_n for positive n. In this case, the “.” denotes the end of the clause. The a, b defines the logical conjunction between literal a and b. This definition of simple logic does not have a logical disjunction. There is no logical disjunction as a binary operator in the clauses, so they are all in conjunctive negation normal form. In the context of a clause, a combination of clauses is comparable to a single clause. The following is the example, (x\_1, x\_2) , (y\_1, y\_2) can be rewritten as x\_1, x\_2, y\_1, y\_2. It is possible for a clause to contain only one literal.

The following are the rules: y :- x\_1, x\_2, ..., x\_n for positive integer n. In this expression, “:-” defines of simple logic specifies the only proof rule, modus ponens. Accordingly, a rule asserts that its consequent follows logically from its antecedent (by means of the inference rule of modus ponens).

The next three logic are statement, knowledge base, and containment. Logic statements are considered complete when they are in the form of rules and clauses. A logical statement ends with - . A knowledge base is a collection of logical statements. The content of a knowledge base is the rules and clauses expressed therein. A literal is contained in its knowledge base if, within it, there exists a clause that asserts it. There is sufficient evidence that a literal is contained by any clause in the set of asserting clauses.

The last logic we use in this component is entailment. The basis of a literal is entailed by its knowledge base if, within it, there is a rule that asserts it as its consequential, and all the antecedent literals of the rule are either contained or entailed by the base. This set of rules shows that the only contained or entailed antecedent literals of any literal asserted (as its consequential) provide sufficient evidence of the literal's entailment.

Figure 5.2 presents the screenshot of the clause in logic.py as the example of the above logic we explained. It is defined by the class in the Python file. In the clause, the literal instances are asserted shown in \_init\_. The clause instance is returned as a string in prolog in the \_str\_(self). We also need hashability in this file.



Figure 5.2 The class of clause in logic.py

* 1. KnowledgeBase

In this knowledge based decision support system, the KnowledgeBase.py is used. This file have the ability to create a simple logic (knowledge base) KB by either the process of creating Literals, Clauses and Rules from a collection of Clauses and Rules, creating sets of corresponding Literals, Clauses and Rules by reading simple logic KB contents from a text file, and the creation of literals, clauses, and rules corresponding to a prolog string representation of a simple logic KB. For every Literal instance in the KB (which it mutually references), generate a Case instance and request the Cases to populate their supports (the attributes associated with the support).

1. Research Methods

Decision influencing factors have always been a problem worth studying, which can be said to be the nightmare of all machine learning practitioners, data analysts, and data scientists. For example, feature selection is often interfered by human factors.

The algorithm of the system in this project mainly deals with errors, missing values, spelling errors and inconsistent values in the collected data through Bayesian machine learning and Kmean algorithm. The objective is to reduce manual modification and intervention factors, complete automatically through the algorithm, and deduce information from common sense knowledge based on probability statistics. The calculated probability distribution provides more information to help the decision-making system, and thus moves away from relying solely on human common sense.

This report is based on python3/numpy/pandas/pgmpy libraries. We used the innate bayesian algorithm and combined with other method to remove unnecessary data, such as giving KNN classification of machine learning to remove duplicate data. We also combine machine learning methods in engineering fusion (sklearn/pgmpy as we introduced before), in order to improve the usability and automation of the project.

6.1 Data Visualization

How do you analyze large, complex and multidimensional data?  The answer is to provide intuitive, interactive, and responsive visualizations like the human eye. Therefore, the main characteristics of data visualization technology are: (1) interactivity.  Users can easily manage and develop data in an interactive manner.  (2) Multidimensional.  You can see multiple properties or variables representing the data of an object or event, and the data can be categorized, sorted, combined, and displayed according to the value of each of its dimensions.  (3) Visibility.  Data can be displayed in images, curves, two-dimensional graphics, three-dimensional objects and animations, and their patterns and relationships can be visually analyzed. Since the computer began to be used in visualization technology, people have found many new visualization technology, the existing technology has been improved, and the application field has been extended to large-scale data set visualization and dynamic interactive display.

In this project, we will use Seaborn as the tool to do visualization. Seaborn is a statistical analysis library for Python that is based on matplotlib and has functions that interact well with pandas.  When we use machine learning methods to analyze and solve problems, we generally follow the process which is related to machine learning.  One important step is exploratory data analysis (EDA).  In this step, the results of the analysis are usually visualized so that they can be observed by themselves and the characteristics of the data can be better described to others.

6.2 Bayesian inference machine learning

It leads to an interesting question in itself to verbalize what we typically treat as mathematical work. What does “B” mean? Since there is uncertainty in the real world as to what 'truth' actually is, it's hard for propositional logic to cope with real-life modeling uncertainties. The noisy nature of independent variables (A) and dependent variables (B) and their invariable non-deterministic nature make it impossible to obtain accuracy measures. As a result, probability theory provides a framework for doing meaningful reasoning in the face of uncertainty that provides a solid basis for meaningful reasoning. The Bayes rule, in particular, provides a "logic of uncertainty" as a by-product of probability theory [17]. Based on a particular value of A, we can estimate the probability that a binary value (say B) will hold by using its conditional probability P(B): that is, “How likely is it that B will occur if A takes the given value?” 0.6 is a good probability for B to be true.

By treating parameter w as a vector consisting of all 'adjustable' parameters in the model, Bayesian analysis treats A and B like random variables. Hence, (B|A, w) denotes the conditional probability, with both B and parameters explicitly dependent on each other. By inferring a distribution over parameters w from the distribution, Bayes' Rule does not optimize a quality measure. Prior to observing the data, there must be a 'prior' distribution over w which we can establish as p(w). Even though it might seem inefficient at first glance, it enables us to recall our assumptions and constraints. We can deal with uncertainties in this way, while staying aware of what we are doing. As a result, simple models with a small amount of detail are usually preferred to complicated ones with too many. In particular, it is important to emphasize that the prior p(w) remains informative even when the property holds. By utilizing a Bayesian approach, we are integrating over a number of variables, which is an intractable problem. Due to this, Bayesian modeling is now the focus of much of the contemporary research.

6.3 K-means algorithm

Before introducing K-means, we will introduce what the clustering is. Clustering is an unsupervised learning that groups similar objects into clusters and groups similar pieces of data together, a bit like automatic categorization. The biggest difference between clustering and classification is that the target of classification is known in advance, while clustering is different and the target of classification is not known. K-means is an unsupervised classical clustering algorithm. It is calculated according to the feature vectors of different objects and then uses the similarity algorithm. The commonly used Euclidean distance, Manhattan distance, cosine distance and so on can be viewed: summary of distance calculation. However, in practical problems, the dimensions of different features are not the same. When transforming into feature vectors, we usually normalize these features respectively before forming their corresponding feature vectors. The selection of similarity algorithm also needs to be made according to the specific scene.

1. means algorithm is a method to construct K partition clusters according to the given data set of N data objects, and each partition cluster is a cluster. This method divides the data into N clusters, each of which has at least one data object, and each data object must belong to and only belong to one cluster. At the same time, the similarity of data objects in the same cluster should be high, while the similarity of data objects in different clusters should be small. Cluster similarity is calculated by means of the mean value of objects in each cluster.  
   The processing flow of k-means algorithm is as follows. Firstly, k data objects are randomly selected, and each data object represents a cluster center, that is, k initial centers are selected. For each remaining object, it is assigned to the cluster corresponding to its most similar cluster center according to its similarity (distance) to each cluster center; The average value of all objects in each cluster is then recalculated as the new cluster center.  
   The above process is repeated until the criterion function converges, that is, the cluster center does not change significantly. The mean square error is usually used as a criterion function, which minimizes the sum of squares of the distances from each point to the center of the nearest cluster.  
   The new cluster center calculation method is to calculate the average value of all objects in the cluster, that is, to calculate the average value of each dimension of all objects respectively, so as to get the center point of the cluster. For example, a cluster includes the following three data object {,4,8 (6), (8,2,2), (4,6,2)}, then this is the center of the cluster (/ 3 (6 + 8 + 4), (4 + 2 + 6) / 3, (8 + 2 + 2) / 3) = (6,4,4). The K-means algorithm uses distance to describe the similarity between two data objects. The distance functions are Ming distance, Euclidean distance, Ma distance and LAN distance, the most commonly used is Euclidean distance.  
    In the K-means algorithm, k represents the number of divided clusters, while means is the core probability of the algorithm -- the center of mass. There are as many clusters as there are centers of mass, and we should divide sample points into clusters. The solution is taking the center of mass closest to the sample point, and puts it in the cluster of the closest center of mass. At this time, what we need to pay attention to is how to calculate the center of mass. Usually, the mean value of each characteristic quantity of each cluster is selected as the center of mass of the cluster.
2. Results and evaluation
   1. PGM programming

Figure 7.1 shows the PGM programming results. The location query table has been presented in this figure, which contains two locations. As this is a prior probability distribution, Bayes' theorem works in theory, and we believe that our result is consistent with the probability distribution.

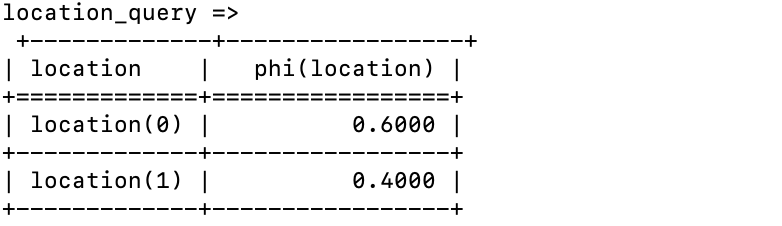


Figure 7.1 PGM programming results

* 1. The inference of model with prior probability

Figure 7.2 shows the prediction information of two selected sentences. We need to predict “I and my wife enjoyed the beautiful room with the wonderful view. The staff was very great the 2 days we stayed.” and “great store, and grocery! Very convenient to the Monument Valley entrance. Staff was friendly, great restrooms, and beautiful views”. The prediction information is given in this figure including the number of cluster and text with the key words. Through manual inspection and verification, we can make our prediction sentence is the most accurate among the candidate sentences.

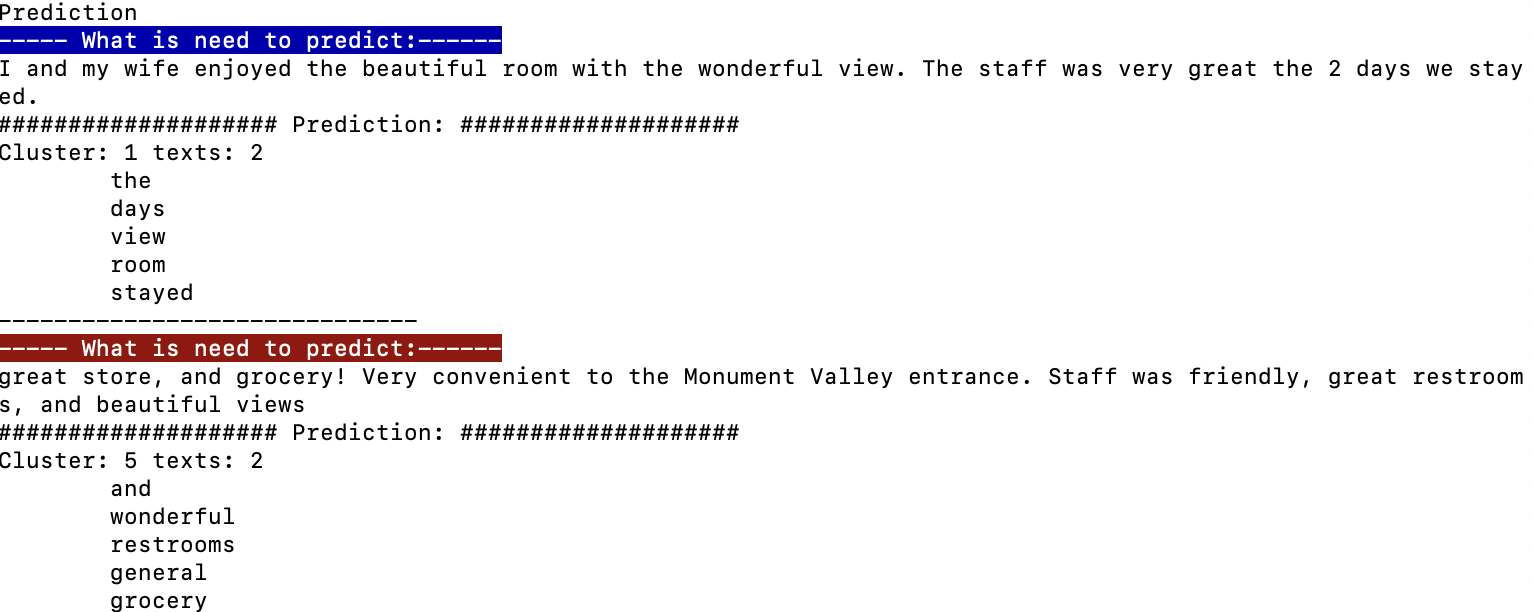


Figure 7.2 The prediction information of two sentences by taking weight based on similarity

1. Conclusion
   1. Summary

In this project, we successfully creating a Argumentative-decision-making system. Through the implementation of Argumentative decision-making, it is easier to connect chaotic and inconsistent data sets to clean records, without the need for manual processing, reducing the interference of human factors in the data, and giving Bayesian probabilistic programming and machine learning help to recomment data. In this way, machine learning and probabilistic programming can provide more human-independent decision factors to decision-making systems, which has potential social benefits: less boring work, less human labor, and the risk of fading away from human experience and common sense.

There are also some risks. This decision-making system, if combined with incomplete information from multiple public sources, could make it cheaper and easier to invade people's privacy, and perhaps even de-anonymize it. The reason is that many probability distributions and knowledge inferences are hidden in data that are difficult for people to analyze individually.

* 1. Prospects and future work

In the future we can try implementing more machine learning methods with probabilistic programming to further reduce human intervention. For example, knowledge graph progression and graph neural network can be considered to further improve the knowledge base in reasoning, de-duplication, clean up dirty data, infer more accurate probability knowledge distribution, and more automatic decision factor analysis.

References **(needed to update)**

1. Wen W, Chen Y H, Chen I C. A knowledge-based decision support system for measuring enterprise performance[J]. Knowledge-Based Systems, 2008, 21(2): 148-163.
2. Eom S B, Lee S M, Kim E B, et al. A survey of decision support system applications (1988–1994)[J]. Journal of the Operational Research Society, 1998, 49(2): 109-120.
3. Bench-Capon T J M, Dunne P E. Argumentation in artificial intelligence[J]. Artificial intelligence, 2007, 171(10-15): 619-641.
4. Argumentation in artificial intelligence[M]. Heidelberg: Springer, 2009.
5. Kukar M, Vračar P, Košir D, et al. AgroDSS: A decision support system for agriculture and farming[J]. Computers and Electronics in Agriculture, 2019, 161: 260-271.
6. Teniwut W, Hasyim C. Decision support system in supply chain: A systematic literature review[J]. Uncertain Supply Chain Management, 2020, 8(1): 131-148.
7. Kumar T S. Data mining based marketing decision support system using hybrid machine learning algorithm[J]. Journal of Artificial Intelligence, 2020, 2(03): 185-193.
8. Keenan P B, Jankowski P. Spatial decision support systems: Three decades on[J]. Decision Support Systems, 2019, 116: 64-76.
9. Lieske S N, Coupal R H, Hamerlinck J D, et al. Planning support systems for fiscally sustainable planning[M]//Planning support systems for sustainable urban development. Springer, Berlin, Heidelberg, 2013: 127-147.
10. Berner E S, Lande T J L. Overview of clinical decision support systems[M]//Clinical decision support systems. Springer, New York, NY, 2007: 3-22.
11. Phillips-Wren G, Daly M, Burstein F. Reconciling business intelligence, analytics and decision support systems: More data, deeper insight[J]. Decision Support Systems, 2021, 146: 113560.
12. Walton D. Argumentation theory: A very short introduction[M]//Argumentation in artificial intelligence. Springer, Boston, MA, 2009: 1-22.
13. Russell B, Whitehead A N. Principia Mathematica Vol. I[J]. 1910.
14. Newton P, Driver R, Osborne J. The place of argumentation in the pedagogy of school science[J]. International Journal of science education, 1999, 21(5): 553-576.
15. Ranney M, Schank P. Toward an integration of the social and the scientific: Observing, modeling, and promoting the explanatory coherence of reasoning[J]. Connectionist models of social reasoning and social behavior, 1998, 245: 274.
16. Bouwman M J. Human diagnostic reasoning by computer: An illustration from financial analysis[J]. Management Science, 1983, 29(6): 653-672.
17. Dubois D, Prade H. Possibility theory, probability theory and multiple-valued logics: A clarification[J]. Annals of mathematics and Artificial Intelligence, 2001, 32(1): 35-66.