KAPITEL 6 / *CHAPTER 6* [28]
CONSTRUCTION OF A FUZZY KNOWLEDGE BASE FOR THE OPTIMIZATION OF AUTOMOBILE FUEL COSTS

DOI: 10.30890/2709-2313.2023-21-01-011

Introduction

Expert systems are computer programs designed to simulate the decision-making abilities of human experts in certain domains. They are a type of artificial intelligence (AI) technology that uses a knowledge base, an inference engine, and a user interface to provide intelligent advice or make decisions based on a set of rules or knowledge.

Key components of the expert system include:

Knowledge Base: This is a repository of specialized knowledge and information related to a particular domain. A knowledge base contains facts, rules, heuristics, and relationships that an expert will use to solve problems or make decisions.

Logical Inference Engine: The logical inference engine is the reasoning component of an expert system. It applies logical rules and algorithms to a knowledge base to make inferences, draw conclusions, and provide recommendations or solutions.

User interface: The user interface provides interaction between the user and the expert system. It can take the form of a text interface, a graphical interface, or even a voice interface that allows users to enter queries, receive responses, and provide feedback.

Expert systems are particularly useful in areas where specialized knowledge is required, such as medicine, finance, engineering, and troubleshooting complex systems. They can help professionals by providing accurate and consistent advice, reducing human error and facilitating knowledge transfer.

Expert systems development involves knowledge acquisition, which is the process of capturing and encoding the knowledge of human experts into a format that can be understood by a computer. This can be done through interviews, consultations and analysis of available resources.

Although expert systems have existed for several decades, their application has evolved along with the development of AI and machine learning. Today, they can be integrated with other technologies such as natural language processing, data analytics, and machine learning algorithms to expand their capabilities and improve decision-making processes.

[28]**Authors:** *Skakalina O.V., Kapiton A.M.*

 ISBN 978-3-949059-94-0

In general, expert systems are valuable tools for gathering and using specialized knowledge, supporting decision-making, and providing intelligent advice in a wide range of domains.

Fuzzy sets

Fuzzy sets are a fundamental concept in fuzzy logic that allows representing and handling uncertainty and uncertainty in data and knowledge. Unlike traditional crisp sets, which have binary membership (an element either belongs to the set or it does not), fuzzy sets assign degrees of membership to elements, representing the degree to which an element belongs to the set[1].

The membership function is used to determine the degree to which an element belongs to a fuzzy set. It maps each element of the universe of discourse (the set of all possible values) to a value from 0 to 1, which indicates the degree to which the element belongs to the set. A value of 1 means full membership and a value of 0 means no membership.

For example, consider the fuzzy set "Temperature", which represents the linguistic term "Hot". A membership function for this fuzzy set can assign a membership degree of 0.8 to a temperature of 30 degrees Celsius, indicating that it is relatively high. A temperature of 20 degrees Celsius might have a membership value of 0.2, indicating that it is not very hot.

Fuzzy sets can have different shapes for their membership functions, including triangular, trapezoidal, Gaussian, and sigmoidal curves, among others. The choice of form depends on the nature of the problem and the area being modeled[2].

Fuzzy sets provide a flexible way to model and represent imprecise or uncertain information. They allow for a gradual transition between membership and non-membership, recording a gradual change in the degree of truth. This enables fuzzy logic to handle situations where there is some degree of ambiguity, uncertainty, or partial information.

Fuzzy sets find applications in various fields, including decision making, control systems, pattern recognition, and data analysis. They are particularly useful in situations where the data or knowledge is inherently fuzzy, vague, or uncertain, allowing for the modeling and analysis of complex systems that do not fit well into neat binary frameworks.

Fuzzy logic

Fuzzy logic is a branch of mathematics and a methodology for dealing with uncertainty and imprecision in decision making. Unlike traditional binary logic, which operates on strict true/false values (0 or 1), fuzzy logic allows degrees of truth, using a range of values from 0 to 1.

In fuzzy logic, variables and operators can have degrees of set membership. Instead of a clear boundary between membership and non-membership, fuzzy logic assigns degrees of truth to determine how well a statement or variable meets a given condition[3]. This enables fuzzy logic to handle situations that involve vagueness, ambiguity, and incomplete information.

Key components of fuzzy logic include:

Fuzzy sets: Fuzzy sets are sets that admit partial membership. Instead of an element being exclusively a member or non-member of a set, it can have a membership degree between 0 and 1. Fuzzy sets are defined by membership functions that describe the membership degree for each element.

Fuzzy Logic Operations: Fuzzy logic uses certain operations to combine fuzzy sets and produce a fuzzy conclusion. The most common operations are fuzzy conjunction (AND), fuzzy disjunction (OR), and fuzzy negation (NOT). These operations consider the membership degree of the elements and produce corresponding fuzzy outputs.

Fuzzy Rules: Fuzzy rules are IF-THEN statements that define relationships between input variables and output variables. They are based on linguistic terms and fuzzy sets. For example, an IF-THEN rule in a temperature control system might be: IF temperature is low, THEN increase heat.

Fuzzy Logic Inference System: A fuzzy logic inference system is a framework that applies fuzzy logic rules to make decisions or inferences based on fuzzy inputs. It uses fuzzy logic operations and rules to process input variables and produce fuzzy output values.

Fuzzy logic has applications in various fields, including control systems, pattern recognition, decision making, and artificial intelligence. This is particularly useful in situations where exact mathematical models are difficult to define or where human experience and intuition play a significant role.

By using fuzzy logic, systems can more efficiently handle real-world scenarios with uncertainty and imprecision. Fuzzy logic provides a flexible and intuitive way to model and solve complex problems that involve subjective or uncertain information.

Expert systems based on fuzzy logic

Fuzzy logic can be integrated into expert systems to enhance their decision-making capabilities by handling uncertainty and imprecision in the knowledge base and inference process. This is how fuzzy logic can be used in expert systems:

Fuzzy representation of knowledge: In traditional expert systems, knowledge is usually represented by clear rules and facts. However, in many fields, knowledge is inherently uncertain or imprecise. Fuzzy logic allows knowledge to be represented using fuzzy rules and fuzzy sets that can handle degrees of truth and uncertainty. This allows experts to express their knowledge in a more flexible and natural way[4].

Fuzzy Inference: Fuzzy logic provides a powerful mechanism for logical inference in expert systems. An inference engine can use fuzzy rules that include fuzzy sets and fuzzy logical operations to make decisions and inferences. Fuzzy inference allows the handling of imprecise or incomplete data, making the expert system more robust and able to deal with real-world scenarios that include uncertainty.

Fuzzy Quantification: Fuzzy logic allows the representation and manipulation of quantitative information that is imprecise or uncertain. Fuzzy quantification techniques, such as fuzzy numbers or fuzzy measures, can be used to work with uncertain data and perform computations in an expert system. This is particularly useful in areas where exact numerical values are difficult to determine or when there is ambiguity in the data[5].

Fuzzy control: Expert systems that incorporate fuzzy logic can be applied to control systems where decisions must be made based on continuous or analog inputs. Fuzzy control systems use fuzzy rules and fuzzy sets to continuously interpret and respond to input data, enabling adaptive and flexible control. This is useful in areas such as process control, robotics and automation.

By integrating fuzzy logic into expert systems, systems become more capable of dealing with uncertainty, inaccuracy, and incomplete information.

6.1. The main stages of building a fuzzy knowledge base

Setting the task of building a fuzzy knowledge base involves creating a system that is able to learn and make decisions based on a large amount of data and expert knowledge[6]. The main goal is to model and represent the knowledge of human experts in a certain field using fuzzy rules and facts.

Building a fuzzy knowledge base includes the following stages:

• *Defining goals:* Defining the main goals and objectives that the fuzzy knowledge base should solve. For example, it can be decision-making, data classification, pattern recognition, etc.

• *Data collection:* Collection of large amounts of data that includes both quantitative and qualitative information. This data can be obtained from various sources, including databases, research, peer reviews, etc.

• *Defining linguistic variables:* Establishing the linguistic variables that will be used to describe the elements of the knowledge base. These may be terms that reflect the vagueness and ambiguity of concepts in the field.

• *Definition of fuzzy rules:* Development of a set of fuzzy rules that reflect relationships between input variables and output results. These rules are based on expert knowledge and experience, and they help make decisions in fuzzy logic.

• *Learning and optimization:* Application of learning and optimization algorithms to adjust the parameters of a fuzzy knowledge base in order to achieve better efficiency and decision-making accuracy.

• *Validation and evaluation:* Verification and evaluation of the effectiveness of the fuzzy knowledge base based on test data or comparison with the results of expert decisions[7]. This allows you to determine how well the system works and meets the set goals.

The resulting fuzzy knowledge base can be used to make decisions in various fields, such as science, medicine, finance, production, and others[8].

6.2. Analysis of the subject area

Entry condition: Build a fuzzy knowledge base (use at least 5 linguistic variables) for the task of calculating gasoline consumption (take into account the type of maneuvers, the level of driver training, the condition of the car, the type of car, etc.), check it for completeness and make a fuzzy conclusion for specific values (choose randomly).

The following variables with five linguistic terms each can be considered to create a fuzzy knowledge base for calculating gasoline consumption, taking into account the type of maneuvers, the level of driver training, the condition of the vehicle, the type of vehicle, etc.:

Shunting:

- *Linguistic terms:* very cautious, cautious, moderate, aggressive, very aggressive;

- *Description*: represents different levels of driving maneuvers, ranging from very soft and economical to very aggressive and fuel consuming.

Driver training level:

- *Linguistic terms*: Inexperienced, Beginner, Intermediate, Experienced, Expert;
- *Description:* Represents different levels of driver training and experience, from novice drivers to highly skilled experts.

Car condition:

• *Linguistic terms:* bad, satisfactory, average, good, excellent;

• *Description:* Displays the overall condition and maintenance of the vehicle, with poor condition indicating high fuel consumption and excellent condition indicating lower fuel consumption.

Vehicle type:

• *Linguistic terms:* compact, sedan, SUV, pickup, luxury;

• *Description:* Represents different types of cars, ranging from compact and economical to luxury and potentially more fuel efficient.

Driving environment:

• *Linguistic terms:* City, Suburban, Highway, Mountain, Mixed

• *Description:* Describes different driving conditions or terrain, such as city driving, suburban driving, highways, mountainous regions, or mixed driving conditions.

By defining the appropriate membership functions and rules based on these linguistic variables, a fuzzy inference system can be built to accurately estimate fuel consumption. The system will take inputs for these variables and provide a fuzzy output that represents the estimated fuel consumption.

6.3. Fragment of the program listing

```
from matplotlib import pyplot as plt
import simpful as sf

def plot_influence(title, range, ling_var, inference_func):
    X = list(range)
    lv = ling_var
```

 ISBN 978-3-949059-94-0

```
    Y = [inference_func(x) for x in X]

    fig, ax = plt.subplots(2, 1)
    ax[0].set_title(title)
    lv.draw(ax[0])
    ax[1].plot(X, Y)

def get_result_output(Maneuvers, TrainingLevel, car_condition, car_type, DrivingEnvironment, fuzzy_system):
    fuzzy_system.set_variable("Maneuvers", Maneuvers)
    fuzzy_system.set_variable("TrainingLevel", TrainingLevel)
    fuzzy_system.set_variable("CarCondition", car_condition)
    fuzzy_system.set_variable("CarType", car_type)
    fuzzy_system.set_variable("DrivingEnvironment", DrivingEnvironment)

    return fuzzy_system.inference()["Consumption"]

def main():
    fuzzy_system = sf.FuzzySystem()

    # Define the fuzzy sets for the linguistic variable "Maneuvers"
    # (В даному випадку - змінна маневри та її терми це та чи інша швидкість на якій автомобіль має
керованість)
    Very_Gentle = sf.TrapezoidFuzzySet(10, 20, 30, 40, term="Very_Gentle")
    Gentle = sf.TrapezoidFuzzySet(30, 40, 50, 60, term="Gentle")
    Moderate = sf.TrapezoidFuzzySet(50, 60, 70, 80, term="Moderate")
    Aggressive = sf.TrapezoidFuzzySet(70, 80, 90, 100, term="Aggressive")
    Very_Aggressive = sf.TrapezoidFuzzySet(85, 110, 120, 130, term="Very_Aggressive")

    Maneuvers = sf.LinguisticVariable([Very_Gentle, Gentle, Moderate, Aggressive, Very_Aggressive],

                              concept="Maneuvers",
                              universe_of_discourse=[10, 130])

    # Define the fuzzy sets for the linguistic variable "TrainingLevel"
    # (В даному випадку це рівень підготовки водія за відсотковою шкалою від 0% до 100% де можна
припустити що 0% людина яка ніколи не
# Define the fuzzy sets for the linguistic variable "DrivingEnvironment"
    Highway = sf.TriangleFuzzySet(0, 0.5, 2, term='Highway')
    Suburban = sf.TriangleFuzzySet(1, 3, 3.5, term='Suburban')
    City = sf.TriangleFuzzySet(3, 4, 5, term='City')
    Mountainous = sf.TriangleFuzzySet(4, 5.5, 6.5, term='Mountainous')
    Mixed = sf.TriangleFuzzySet(5.5, 6, 7, term='Mixed')

    DrivingEnvironment = sf.LinguisticVariable([Highway, Suburban, City, Mountainous, Mixed],
                            concept="DrivingEnvironment",
                            universe_of_discourse=[0, 7])
```


ISBN 978-3-949059-94-0

```
# Add the linguistic variables to FuzzySystem
fuzzy_system.add_linguistic_variable("Maneuvers", Maneuvers)
fuzzy_system.add_linguistic_variable("TrainingLevel", TrainingLevel)
fuzzy_system.add_linguistic_variable("CarType", CarType)
fuzzy_system.add_linguistic_variable("CarCondition", CarCondition)
fuzzy_system.add_linguistic_variable("DrivingEnvironment", DrivingEnvironment)

 # Define the fuzzy sets for the linguistic variable "Consumption"
Consumption = sf.AutoTriangle(4,
                    terms=["Low", "Moderate", "High", "Very High"],
                    universe_of_discourse=[0.5, 30])
fuzzy_system.add_linguistic_variable("Consumption", Consumption)

# Define the fuzzy rules
fuzzy_system.add_rules([
   # Maneuvers
   "IF (Maneuvers IS Very_Gentle) THEN (Consumption IS Low)",
   "IF (Maneuvers IS Gentle) THEN (Consumption IS Moderate)",
   "IF (Maneuvers IS Moderate) THEN (Consumption IS Moderate)",
   "IF (Maneuvers IS Aggressive) THEN (Consumption IS High)",
   "IF (Maneuvers IS Very_Aggressive) THEN (Consumption IS Very_High)",
   # # TrainingLevel
   "IF (TrainingLevel IS Inexperienced) THEN (Consumption IS Very High)",
   "IF (TrainingLevel IS Beginner) THEN (Consumption IS High)",
   "IF (TrainingLevel IS Intermediate) THEN (Consumption IS Moderate)",
   "IF (TrainingLevel IS Experienced) THEN (Consumption IS Low)",
   "IF (TrainingLevel IS Advanced) THEN (Consumption IS Low)",
   # CarCondition
   "IF (CarCondition IS Poor) THEN (Consumption IS Very_High)",
   "IF (CarCondition IS Fair) THEN (Consumption IS High)",
   "IF (CarCondition IS Average) THEN (Consumption IS Moderate)",
   "IF (CarCondition IS Good) THEN (Consumption IS Moderate)",
   "IF (CarCondition IS Excellent) THEN (Consumption IS Low)",
   # # Car_type
   "IF (CarType IS Compact) THEN (Consumption IS Low)",
   "IF (CarType IS Sedan) THEN (Consumption IS Low)",
   "IF (CarType IS SupportUtility) THEN (Consumption IS Moderate)",
   "IF (CarType IS Pickup) THEN (Consumption IS High)",
   "IF (CarType IS Luxury) THEN (Consumption IS Very_High)",
   # DrivingEnvironment
   "IF (DrivingEnvironment IS Highway) THEN (Consumption IS Low)",
   "IF (DrivingEnvironment IS Suburban) THEN (Consumption IS Moderate)",
   "IF (DrivingEnvironment IS City) THEN (Consumption IS High)",
   "IF (DrivingEnvironment IS Mountainous) THEN (Consumption IS Very_High)",
   "IF (DrivingEnvironment IS Mixed) THEN (Consumption IS High)",
```


ISBN 978-3-949059-94-0

```
])

while True:
    maveuvers_input = input("Введіть швидкість при якій здійснюється маневр (0-150км/год): ")
    try:
        maveuvers_input = int(maveuvers_input)
        if 0 <= maveuvers_input <= 150:
            break
    except Exception:
        pass

while True:
    training_level_input = input("Введіть рівень навичок (0 - 100%): ")
    try:
        training_level_input = int(training_level_input)
        if 0 <= training_level_input <= 100:
            break
    except Exception:
        pass

while True:
    car_condition_input = input("Введіть стан авто (0-100%): ")
    try:
        car_condition_input = int(car_condition_input)
        if 0 <= car_condition_input <= 100:
            break
    except Exception:
        pass

while True:
    car_type_input = input("Введіть тип авто (0.8-5.0л): ")
    try:
        car_type_input = float(car_type_input)
        if 0 <= car_type_input <= 5:
            break
    except Exception:
        pass

while True:
    environment_input = input("Введіть складність оточення водія (0-7): ")
    try:
        environment_input = float(environment_input)
        if 0 <= environment_input <= 7:
            break
    except Exception:
        pass
```


ISBN 978-3-949059-94-0

```
    plot_influence("Maneuvers", range(10, 130, 20), Maneuvers,
                   lambda x: get_result_output(x, 80, 80, 1.3, 1.0, fuzzy_system))
    Maneuvers.plot()

    plot_influence("Training Level", range(0, 100, 10), TrainingLevel,
                   lambda x: get_result_output(60, x, 80, 1.3, 1.0, fuzzy_system))
    TrainingLevel.plot()

    plot_influence("Car Condition", range(0, 100, 10), CarCondition,
                   lambda x: get_result_output(60, 80, x, 1.3, 1.0, fuzzy_system))
    CarCondition.plot()

    plot_influence("CarType", range(0, 5, 1), CarType,
                   lambda x: get_result_output(60, 80, 80, x, 1.0, fuzzy_system))
    CarType.plot()

    plot_influence("Driving Environment", range(0, 7, 1), DrivingEnvironment,
                   lambda x: get_result_output(60, 80, 75, 1.3, x, fuzzy_system))
    DrivingEnvironment.plot()

    Consumption.plot()

    output_result = get_result_output(maveuvers_input,
                                      training_level_input,
                                      car_condition_input,
                                      car_type_input,
                                      environment_input,
                                      fuzzy_system)

    print(f"Витрата палива: {output_result}")

if __name__ == "__main__":
```

6.4. Results of building a fuzzy knowledge base

To perform testing of the developed expert system, we will set several sets of input parameters and record the results of the obtained output variable, which is responsible for fuel consumption in certain conditions, in certain types of cars, in certain conditions and at certain levels of driver skills.

In this case, 3 cases were set for calculating fuel consumption. The results of calculations are shown in Figure 4.1. Figures 4.2-4.6 show graphs of fuzzy given sets of linguistic variables.

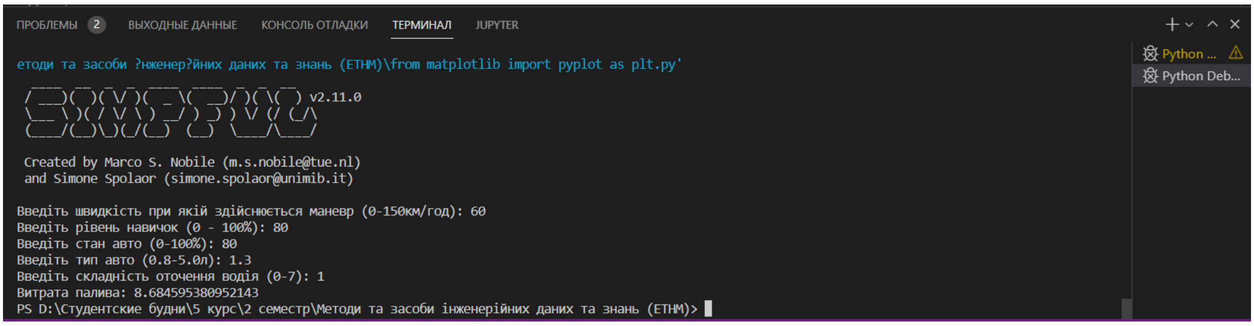

Figure 1 - Fuel consumption for a driver with a speed of 60 (average level of maneuvers), high experience and a small car on the highway

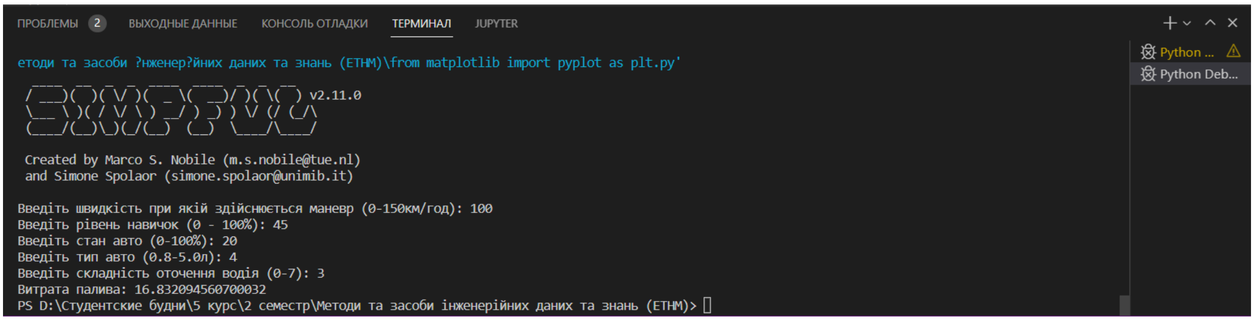

Figure 2 - Fuel consumption for a driver with a speed of 100 (aggressive level of maneuvers), little experience and a powerful car in the city

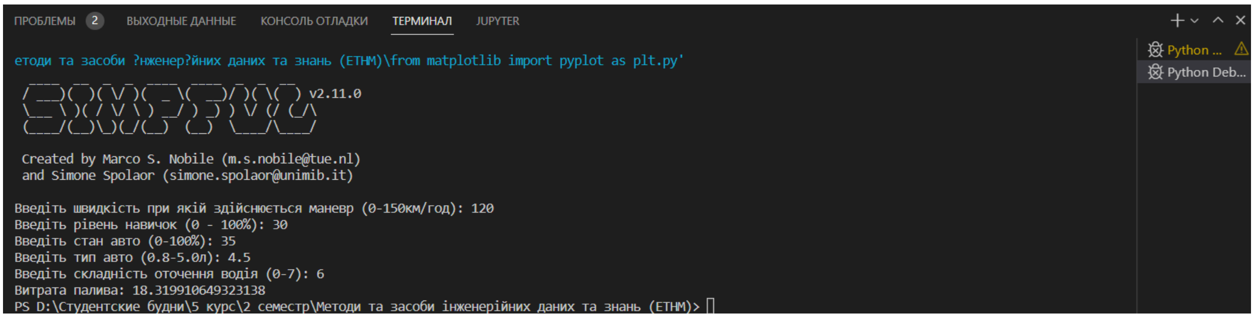

Figure 3 - Fuel consumption for a driver with a speed of 120 (very aggressive level of maneuvers), little experience and a very powerful car in mixed conditions (city + mountain road)

Also, during system testing, two types of graphs were created for each input variable. The first type depicts the membership function of a variable, and the second - the dependence of an output variable on one input, provided that all other values are fixed. The second type of graphs is especially useful, because it helps to understand the nature of the influence of the input variable on the expected result and to verify the correctness of the created rules of the fuzzy knowledge base.

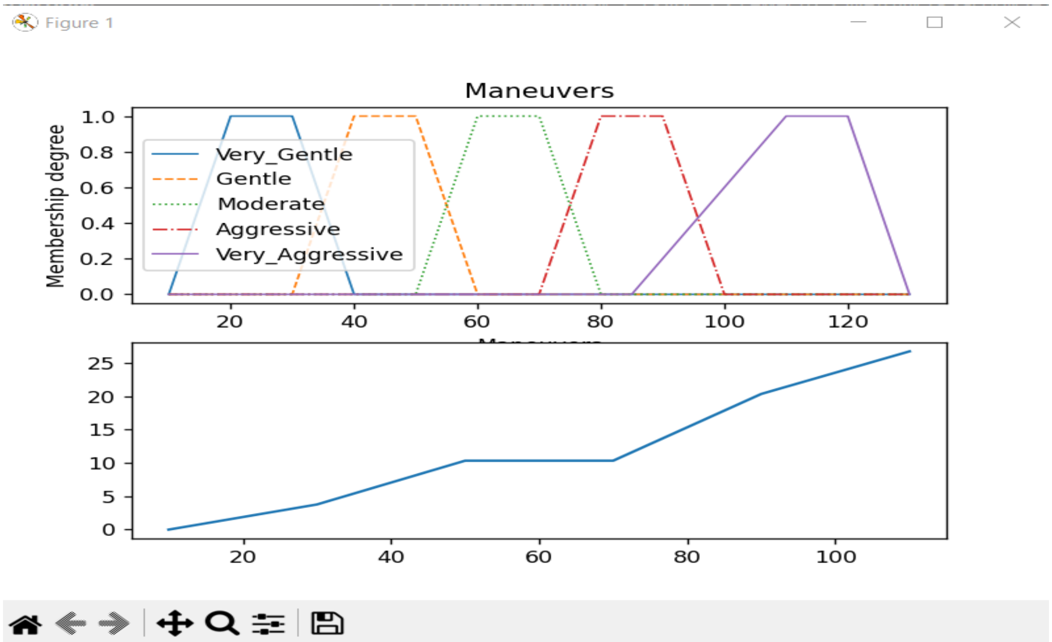

Figure 4 - "Maneuvers" variable. The graph of the variable membership function and the dependence of the output variable

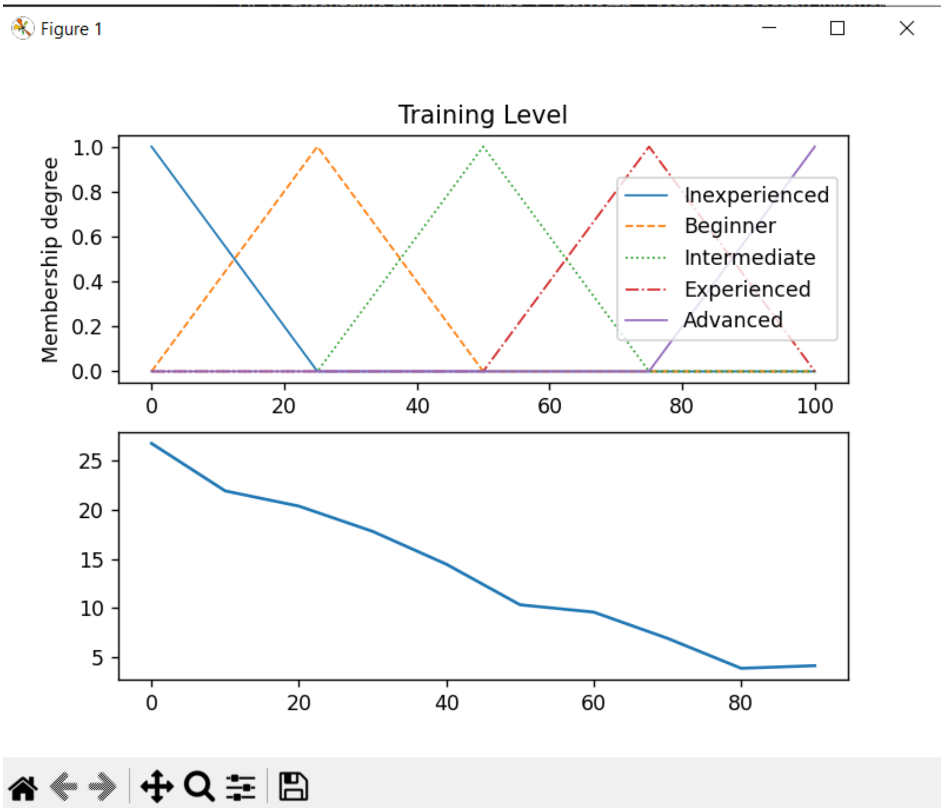

Figure 5 - Variable "Level of driver training". The graph of the variable membership function and the dependence of the output variable

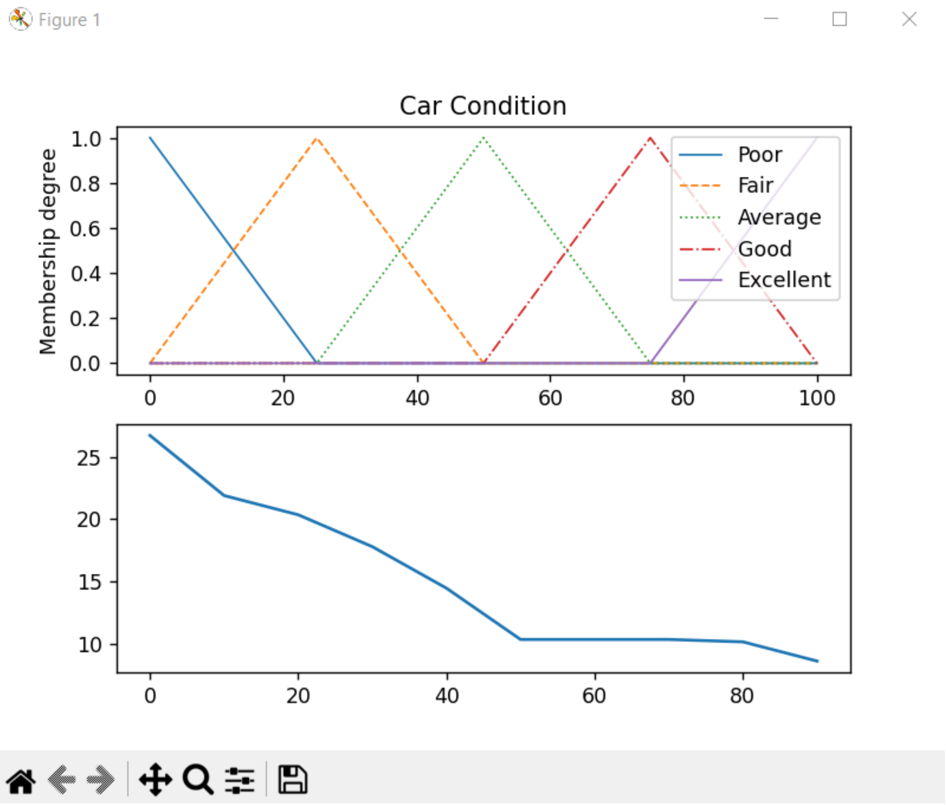

Figure 6 - Variable "Car State". The graph of the variable membership function and the dependence of the output variable

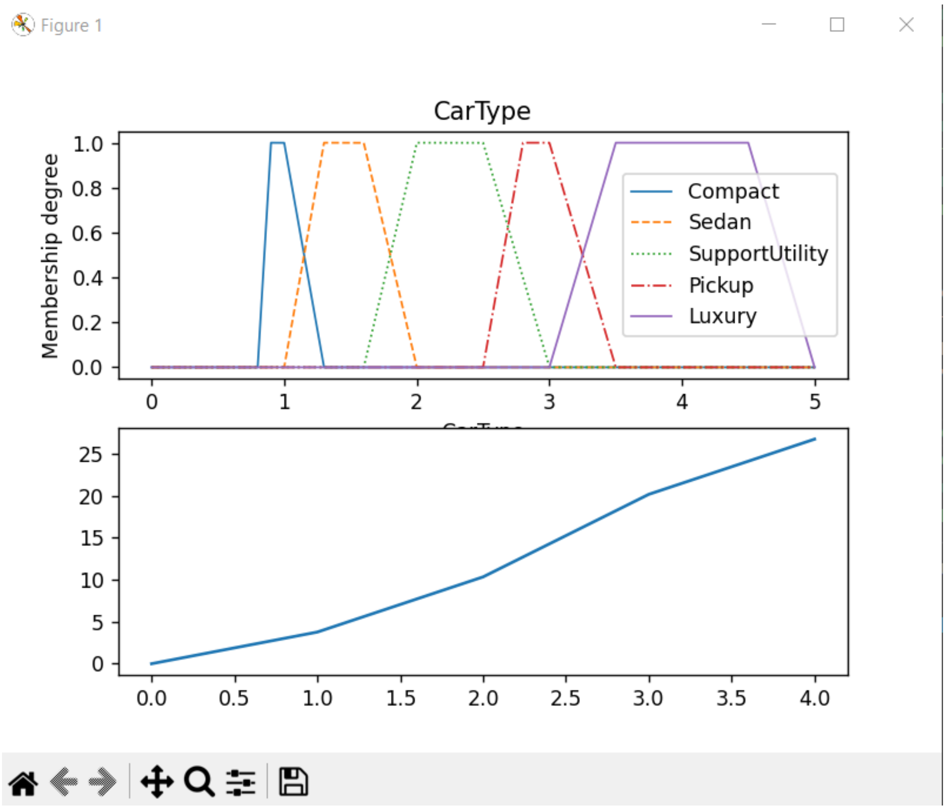

Figure 7 - Variable "Car State". The graph of the variable membership function and the dependence of the output variable

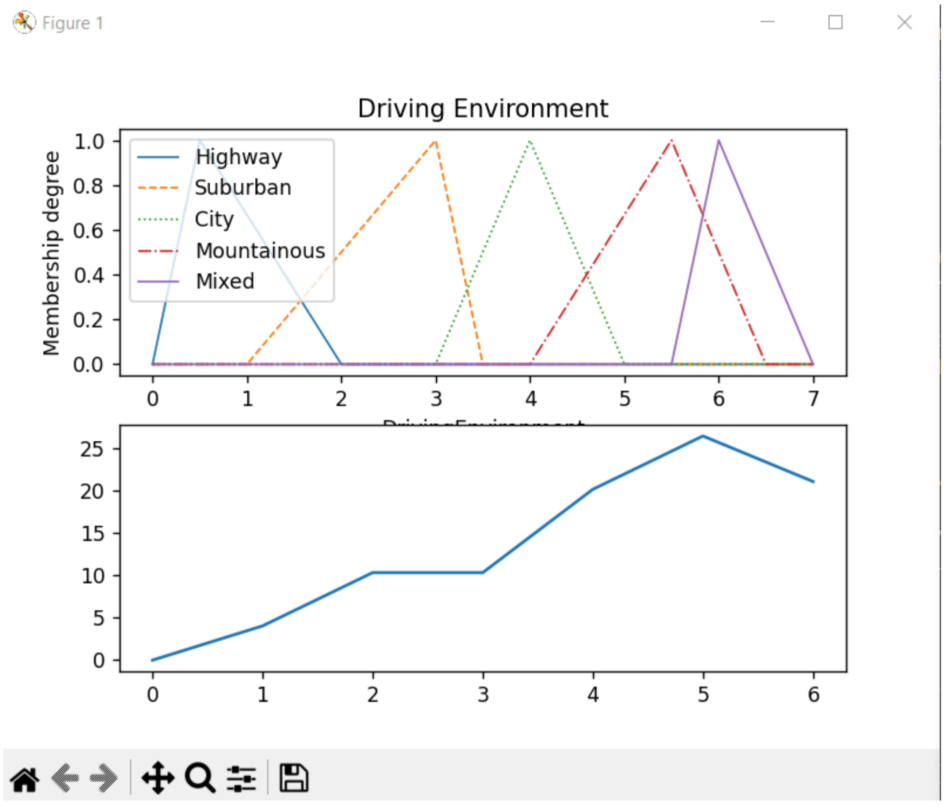

Figure 8 - Variable "Driving environment". The graph of the variable membership function and the dependence of the output variable

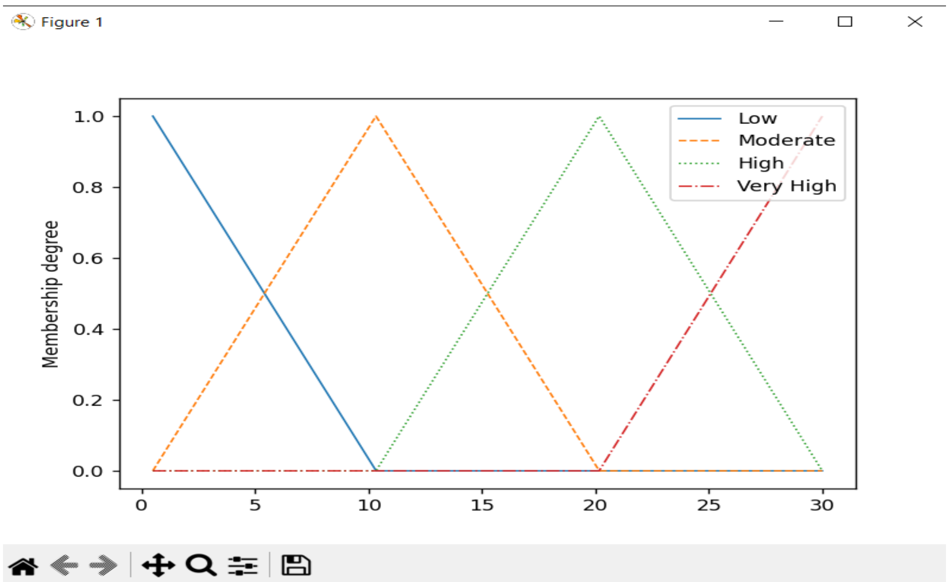

Figure 9 - "Fuel consumption" variable. The graph of the membership function of a variable

 ISBN 978-3-949059-94-0

Conclusions

A fuzzy knowledge base has been created for calculating fuel consumption. Five linguistic variables, each containing five terms, are considered to capture different aspects that may affect fuel consumption: type of maneuvers, driver training level, vehicle condition, vehicle type, and driving environment.

Through the application of fuzzy logic, the complexities and uncertainties associated with fuel consumption have been fully taken into account. By defining linguistic terms and membership functions, we can accurately model the inputs and outputs of the system. The fuzzy rules we set up allow us to make reasonable estimates of fuel consumption based on given input values.

This fuzzy system offers practical applications for optimizing fuel efficiency and providing valuable information to drivers and car manufacturers. It can help drivers adopt more fuel-efficient driving behaviors, shift vehicle manufacturers' vehicle designs toward improved fuel economy, and help individuals and companies make informed subjective decisions about fuel consumption and costs.

The development of this fuzzy system is an important step towards smarter and more adaptive approaches to managing and reducing gasoline consumption. By considering real-world factors and linguistic variables, we can obtain more accurate and personalized assessments, leading to more efficient and sustainable transportation practices.