

# Introduction

In recent years, artificial intelligence has grown significantly, empowering various industries with expert systems capable of supporting specialized domains. Vehicles can present challenges when issues arise, especially for individuals without technical knowledge in mechanics. This project aims to apply AI principles within automotive diagnostics, making technical troubleshooting more accessible to the general public. We propose developing a medium-complexity expert system in the form of a car troubleshooting chatbot. This chatbot will interact with users to understand symptoms and suggest probable causes and solutions, ultimately enhancing user confidence in basic car maintenance and reducing stress related to automotive issues.

## Problem Statement and Objectives

### Problem Statement:

Car owners frequently encounter vehicle issues that disrupt their schedules and increase expenses. Diagnosing these problems can be challenging without specialized knowledge, often leading to unnecessary stress and costly repairs. With a reliable, easy-to-use tool for troubleshooting, users could save time, money, and reduce stress by addressing minor issues independently. This project focuses on developing an expert chatbot system to assist users in diagnosing common car issues through a structured, conversational interface.

### Objectives:

1. **Develop a car troubleshooting chatbot:** Use Python and Experta to create a chatbot that helps users identify potential car issues based on symptoms provided.
2. **Implement probabilistic reasoning with Bayesian networks:** Integrate pgmpy for Bayesian analysis to improve decision-making accuracy.
3. **Enhance user accessibility to automotive diagnostics:** Provide users with reliable guidance to make informed decisions regarding minor repairs, thereby reducing dependence on expert mechanics for basic troubleshooting.

## Requirements Analysis

For the successful development of the car troubleshooting chatbot, several requirements were identified:

1. **Scope Definition:** The chatbot will address a predefined set of common car issues, including engine problems, brake malfunctions, electrical failures, and other relevant categories.
2. **User Needs:** The chatbot should be intuitive and conversational, catering to users without extensive automotive knowledge. It should provide clear guidance to help them resolve minor car issues independently.
3. **Interface Requirements:** The chatbot will require a user-friendly interface, which allows for easy symptom input and returns diagnostic results promptly.
4. **System Requirements:** The project will utilize the Experta library for rule-based expert systems and pgmpy for Bayesian networks, integrating both in Python.
5. **Database Requirements:** To support personalized user interactions, the system will need a database to store user profiles, interaction logs, and relevant diagnostic information.

## Knowledge Acquisition and Representation

To accurately diagnose car issues, the chatbot will rely on domain-specific knowledge acquired from automotive experts, technical manuals, and standardized troubleshooting resources. The following steps outline the knowledge acquisition and representation process:

1. **Consultation with Automotive Experts:** Mechanics and technicians provide essential insights, defining key symptoms and diagnostic guidelines for various car systems.
2. **Compilation of a Knowledge Base:** Information from technical resources and repair manuals helps establish a standardized knowledge base. This base will include common symptoms, potential causes, and recommended solutions for different car systems.
3. **Development of Bayesian Networks:** Bayesian networks model the probabilistic relationships between symptoms and diagnoses, supporting the system in making informed recommendations based on user input.
4. **Knowledge Representation:** The acquired knowledge will be structured as decision rules within Experta and probabilistic models within pgmpy, enabling the chatbot to reason about complex scenarios and provide recommendations based on the likelihood of each possible issue.

