UNIVERSITY OF NAIROBI

DEPARTMENT OF COMPUTING AND INFORMATICS

A MIXED STRATEGY FOR VEHICLE VALUATION

BY

CHARLES MUTINDA KIILU

A RESEARCH PROJECT REPORT SUBMITTED IN PARTIAL FULFILLMENT FOR THE REQUIREMENTS OF AN AWARD OF THE DEGREE OF MASTER OF SCIENCE IN COMPUTATIONAL INTELLIGENCE OF THE UNIVERSITY OF NAIROBI.

Declaration

Student declaration

This study is an original work and has not been presented in any institutions for any academic award to the best of my knowledge.

Signature ......KMutinda......   Date ......23/08/2021......

Charles Mutinda Kiilu,
P52/6510/2017

Approval of the university supervisor

This project report has been presented for evaluation with my consent as the university supervisor.

Signature   Date 23/08/2021

Prof. Elisha T. O. Opiyo,
Associate Professor of Intelligent Systems,
Department of Computing and Informatics,
University of Nairobi.

Acknowledgement

I would like to express profound gratitude to my beloved family and friends, for the understanding and motivation they have given me during this research, and especially to my mother, Teresia Mutinda, for all the sacrifices she has continuously made so that I could one day have this opportunity. My appreciation extends to my supervisor, Prof. Elisha Opiyo, whose guidance has enabled me to get this far with the research, and his fellow panellists for their insights on how I could improve my work.

Abstract

The number of vehicles in Kenya grows at a rate of 12% annually, with the national registered fleet standing at 4 million as of 2018. All these vehicles have to be valued regularly for a variety of reasons not limited to insurance, resale, leasing and accounting. As such, it is important to have an easy to use, reliable, readily available system that can determine the value of a vehicle given some properties about the said vehicle. The variation of values obtained from different valuers for identical vehicles exposes irregularities in the contemporary automobile valuation systems. When in need of quick car valuation services, the lack of consistent, accurate and readily available tools to perform the required valuation is glaring, as the primary way to get an automobile valued is through contacting an expert from a licensed evaluation firm or an insurance agent. The existing car valuation mechanisms rely chiefly on expert opinions and the use of the formulae to calculate a used car's compound annual depreciation which is subtracted from the price at 0 mileage, adjusted for inflation over the years. There have been attempts to automate vehicle valuation by use of machine learning, which yielded promising results. Multiple regression analysis has been employed to identify vehicle properties that have the greatest bearing on the value of the vehicle, as well as predict the price given values of the different parameters. This approach has also been applied successfully in other domains for valuation of assets such as land and FMCGs. For this study, a multi-agent systems architecture was employed to encapsulate three regression models for vehicle value prediction, as well as a natural language processing model to extract vehicle features from vehicle descriptions in unstructured text. The three models were built and trained to generate predictions, each leveraging either of the SVM-based regression and Neural Networks (ANNs) implementation in WEKA, or the Deep Learning regression provided by WekaDeeplearning4j version 3.8.5. The best performing model provided a reliable option for vehicle valuation, with 11% relative mean error, having been trained on only 1000 rows of data, out of a possible 200,000 records, and thus was used in the design of the functional prototype. Given the temporal, budgetary and computational resource restrictions on this study, there is great potential for improving the performance of the prediction models given more time, data and computing power.

List of important Abbreviations

AMS – JADE platform's in-built Agent Management System

DF – JADE platform’s Directory Facilitator

RMS – JADE platform's in-built Remote Management System

ANNs – Artificial Neural Networks

CUDA – NVIDIA's Compute Unified Device Architecture

GPU – Graphical processing unit

JADE – Open-source Java Agent DEvelopment Framework.

JDK – Java Development Kit, Standard edition

MAS – Multi agent system

MLP – Multi-layer Perceptron

NLU – Natural language understanding

OpenNLP – Apache's open-source OpenNLP project for natural language processing

RSE – Relative Squared Error

SMOreg/SVMreg – SVM-based regression implementation

WEKA - Waikato Environment for Knowledge Analysis

Table of contents

List of Tables and Diagrams

List of Figures

List of Tables

Definition of Important Terms

Autonomous: Capable of acting independently and of exercising control over an internal state.
Semantics: Using NLP techniques to extract meaning from text
Social agent: An agent with the ability to interact with other agents
Software agent: In the context of this research, a software agent (or an agent) is an intelligent autonomous social agent existing within a software multi-agent environment tasked with a particular function.
SMOreg: Regression algorithm based on the scalable vector machine algorithm.

Chapter 1: Introduction

1.1 Background of the study

A report published by the United Nations Environment Programme in 2020, the Used Vehicles and Environment report, stated that second-hand vehicle sales account for nearly 95% of vehicles imported into Kenya annually. As of 2018, there were 3,280,934 registered units as reported by the Kenya National Bureau of Statistics. This accounted for an annual growth rate of 12% as cited in a 2016 report.

These figures suggest that hundreds of thousands of cars are added annually to the existing millions of other old cars operating within Kenya. Prospective buyers, sellers, insurance firms, property valuers, and numerous stakeholders must estimate the value of these cars with the utmost precision.

According to the Automobile Association of Kenya reasons for vehicle valuation include:

Pre-insurance valuation: this valuation is conducted to ensure that vehicular owners pay precise premiums when subscribing to insurance policies for their vehicle and that Insurers generate concise premium rates and minimise revenue loss as a result of value miscalculation.

Technical Brief Valuation: This refers to the precise determination of the price of a car or a machine when purchasing, vending and or disposing of antiquated equipment or in a scenario where an automobile is presented as a guarantee during loan acquisition.

Full Mechanical Valuation: It is an evaluation of a preventative nature and done to identify possible mechanical faults and reduce the chance of a vehicle breakdown.

Pre-Theft/Pre-fire valuation: when a vehicle is stolen this procedure must be conducted by the insuring body. To correctly compensate the owner, the value of the vehicle before it was stolen has to be calculated.

Accident Assessment: This inspection is carried out upon the loss of the vehicle after the loss and a claim settlement have been issued. In circumstances whereby it is difficult to fix the vehicle due to high replacement costs or extensive damage; advice is given on the restorative and salvage cost values.

Examining of automotive components: This is performed when an unbiased perspective is consorted regarding automotive failure. Failure could encompass accidents or automotive damage occurring during a repair.

According to (Bennett, 2016), the globally accepted method used to evaluate the value of an old car consists of pinpointing the vehicle’s value when it was new - and deducting the amortised cost by the cumulative years the vehicle has been used.

The resale value of a vehicle is affected by other factors such as:

Mileage: This refers to the distance covered by a vehicle in miles; the more miles covered lower the car’s worth as it increases its depreciation.

Vehicle Condition: Refers to any sort of damage to both the exterior and interior of a used vehicle; this naturally has a negative impact on the vehicle’s value.

Make and Model: Understandably, vehicles that offer lower fuel consumption over the distance covered fetch higher resale prices are generally popular.

The pricing and availability of spare parts, servicing and models, which are no longer produced, are factors that determine the resale value.

Vehicle Age: Most automobile’s age is proportional to its sale price, the older the vehicle the lower the resale price. This is because vehicles have a lifespan and their functional usage is expected to erode with time.

Ownership chain: The more times an automobile is sold the more its sale price keeps depreciating as a consequence of increasing maintenance costs by latter owners.

After-Sale Service: The quality of post-sale services varies from brand, thus affecting the condition of an automobile over time.

Features, and Options: The availability of options in the market features a car has, the likelier the price is to deteriorate; however, it has been noted that additional safety features increase a car’s value.

Colour of the car: Basic colours such as silver, black, or white have a preference among potential purchasers. Selling a uniquely coloured car increases the difficulty of finding potential customers thus the value of the automobile could be negatively impacted.

Despite numerous pre-existing vehicle valuation mechanisms, value determination is still a major challenge in Kenya. Similar car models can fetch exceedingly different prices from valuers. Concise mechanisms are yet to be installed to ensure that there are standardised vehicular valuation tools to minimise inconsistencies from being experienced,

Stakeholders use different methods and data to inform their valuation processes hence yielding different results, making it hard to get a standard market price for a vehicle. Thus, the importance of having a readily available accurate and consistent vehicle valuation mechanism that ensures the automobile’s worth can be accurately determined.

Natural language processing, artificial neural network algorithms and multi-agent systems are some of the proposed tools that will be used to accurately generate the value of a car.

1.2 Statement of the problem

The existence of irregularities in the contemporary vehicle valuation processes is proven in the variation of values provided by different valuers, for similar vehicles. When you need to determine an acceptable value for a vehicle, the lack of an easily accessible and accurate instrument for vehicle valuation is clear, since the only method to acquire a vehicle assessment is to contact a valuer, an insurance broker or a valuation firm. And without a method to cross-validate the data presented, this could result in incorrect vehicle valuation and potentially inflated vehicle costs and insurance premiums being paid by unwary individuals. Again, the majority of people lack the technical expertise required to construct, grasp, and apply valuation formulae to arrive at car value estimations. (Kieti, 2005) .

1.3 Study Objectives

1.3.1 General Objective

Creating a mixed strategy vehicle valuation prototype that is simple to use, multi-agent-based, and fully functional, and that uses neural networks for numerical regression and natural language processing for feature extraction to forecast the value of a vehicle given precisely defined parameters or textual descriptions.

1.3.2 Specific Objectives

- To aggregate and analyse data using the existing vehicle valuation models from varied domains, and identifying areas with inconsistencies and discrepancies within the current vehicle valuation procedures.
- To design a mixed strategy prototype that considers the findings of the analysis and the deduction on related topics by other researchers.
- To implement a functional mixed-strategy vehicle valuation prototype.
- To amass data and examine the reliability of the evaluation prototype by establishing its performance on real vehicle data.

1.4 Goals

The study aims to produce an accurate easy-to-use vehicle valuation tool for public use, as well as industrial and professional functions such as insurance and car sales.

1.5 Limitations of the study

The accuracy of the models created in this research are subject to inconsistency taking into consideration the varying prices from different car dealers will not be similar for identical cars. Car dealers have different profit and cost considerations and hence produce prices that are expected to differ and compromise the predictions of the real value.

Automobile valuation statistics provided by different dealers are based on different valuation techniques leading to different degrees of accuracy.

Supply and demand, monetary inflation and other financial and market factors have an impact on the valuation of a vehicle. When the market preference is in high demand for a specific model, its value tends to be greater than that of a less preferred model, and these dependent factors will be reflected in the dataset available for this research.

Import taxes on foreign vehicles affect the buying and selling prices, which proves challenging when it comes to accurately identifying the intrinsic value of the vehicle.

1.6 Scope of the study

The scope of this report is confined to the value estimation of cars in Kenya. This study aims to produce a correct vehicle and machinery valuation model that could be implemented in different fields of vehicle valuation such as inspection of motor vehicles, pre-insurance valuation, accident assessment, pre-fire and pre-theft assessments, full mechanical assessment and technical brief assessment.

1.7 Expected Contributions

1) At the closing of this study, it is expected to have derived in an operational automobile valuation framework based on natural language processing, regression algorithms and multi-agent systems to determine the value of a vehicle.

The framework will be expected to serve as a readily available instrument providing a closely accurate market value for an automobile taking into account various characteristics of the said vehicle.

2) Finally, this study aims to direct future research by laying a solid platform for future researchers to build on.

1.8 Proposal Organisation

Chapter 1: Introduction – Describes the definition of the problem, scope, the study's objectives, research questions, and constraints.

Chapter 2: Literature review - This section contains general facts about the relevant work, suggested solution's design and the problem domain.

Chapter 3: Methodology - This chapter covers the whole study process, as well as specific methods to be used, as well as a well-planned timeline, budget, and resource requirements.

Chapter 4: Data Analysis, Prototype Design and Implementation - This segment presents the analysis, design, and execution of the proposed solution.

Chapter 5: Results and Discussion – In this segment, the outcomes of this research are presented and analysed.

Chapter 6: Conclusion and Recommendation - This chapter offers the opinions of the researcher and provides suggestions for the preceding work.

2. Literature Review

2.1 Theoretical literature review

2.1.1 Introduction

This study focuses on the but is not limited to following fields of study; Natural Language Processing, Multi-Agent Systems and Artificial Neural Networks.

The following chapter will explore the above areas as well as other works done previously that inform this study.

2.1.2 Linear Regression

This is a supervised algorithm used in machine learning where statistical approaches are used to make predictions. Vehicle prices, a person's age and employees' salaries that contain numerical variables can be predicted. A linear regression approach shows a linear relationship with one or even more independent variables and a dependent variable (y) (x) (Bonaccorso, 2017).

Linear regression can be divided into two types:
Simple linear regression: A dependent variable's numerical value is predicted by one independent variable. e.g. ***y = mx + c***; the value of y is affected only by the value of x, with m and c being constants.
Multiple linear regression: A dependent variable's numerical value is predicted by many independent factors. e.g. ***y = mx + nw + oz + c***

2.1.3 Natural Language Processing

Natural Language Processing is a subfield of linguistics that involves machine's having the ability to read, decipher, understand and interpret human language. Machine learning aids computers to understand the human language (Bennett, 2016).

Natural Language Processing involves syntactic and semantic analysis.

Semantic Analysis: This refers to the extraction of meaning from text. To comprehend the interpretation of words and sentence structuring, computer algorithms are used. Techniques used in semantic analysis include:

- Natural language generation: Structured data is converted to human language.
- Word sense disambiguation: This is an automatic process that is used to identify the aspect in which a specific word is being and its context. This is because a specific word could have different meanings when used in different contexts.
- Named entity recognition (NER): This involves the identification of text segments to be organized into pre-defined groups e.g., proper names of animals and cities.

Syntax Analysis: It entails examining words in a phrase to verify that they are comprehensible and that their order is logical. Syntactic analysis evaluates how natural language follows grammatical principles in Natural Language Processing. Computer algorithms are employed to determine the grammatical rules that give meaning to a group of words. Syntax techniques include the following:

Morphological segmentation: The act of collapsing words into smaller bits is referred to as morphemes.

Lemmatization: This is the practice of distilling a term down towards its dictionary root. It's more complex than stemming and necessitates a more in-depth knowledge of the language.

Word segmentation: This is the process of breaking down huge chunks of material into smaller, more relevant parts like subjects or words.

The process of classifying parts of the text (corpus) into distinct parts of speech is known as part-of-speech tagging (POS tag). This is determined by the word's definition and the contexts where it is used. POS tag is used in in-text analytics tools and algorithms, as well as corpus searches. A language's foundation is made up of verbs, adverbs, adjectives, nouns, and other speech components. The goal of POS is to tag (assign) the various portions of a speech to a text.

Sentence breaking: This is the method of breaking a vast volume of text into sentences and figuring out where they start and end.

Parsing: This is the method of assessing words in a sentence grammatically.

Stemming: This is a grammatical analysis technique for analysing words in a sentence.

Stemming algorithms, for example, might be used to create a root word - ask - from phrases like ask, asking, and asked.

2.1.4 Agents

An agent is defined by (M. Wooldridge and N.R. Jennings, 1995) as a software system that is located in a contained territory and capable of autonomous operations to achieve its goals.

An agent, according to Yu Mon Zaw and Nay Min Tun (2014), is a computational entity that works on behalf of some other entity (or entities) to complete a job or achieve a goal.

Agents have a number of distinguishing characteristics, including:

- **Autonomy:** They have some independence and so perform certain tasks independently.
- **Intelligence:** With practice and time, agents can enhance their performance at a specific activity. They increase their future performance by learning from their past encounters with data in some contexts.
- **Collaboration:** Agents can communicate with one another, exchange information via messages, issue and receive instructions and assessments, respond to communications, and coordinate with others to accomplish their objectives.
- **Negotiation:** The capability of conducting structured dialogues known as negotiations

facilitates collaborations with other agents.

- **Pro-activity:** Agents can act on internal states and make decisions based on them.
- **Reactivity:** Both external events and stimuli can cause agents to react. To complete their tasks, they modify their behaviour and decisions.

2.1.4.1 Multi-Agent Systems

An agent-based system, according to M. Wooldridge and N.R. Jennings (1995), employs the key abstraction of an agent. Agents are the main components used in the design and implementation of the system. Agent systems are computer programmes that work freely, autonomously, and intelligently to complete certain tasks and achieve specific goals, both alone and as part of a socially enabled team. Agents have a collection of characteristics that distinguish them.

A Multi-Agent System is a system with numerous agents that communicate in a peer-to-peer fashion. There are a variety of commercial and open-source software tools available to help construct computer systems that represent agents, as well as multi-agent systems that model societies of collaborating agents. Two of the most prominent are Java's JADE Framework and Python's SPADE module.

2.1.5 Machine Learning

Machine learning is a branch of computer science that facilitates the creation of analytical models. It is considered to be a component of artificial intelligence.
With minimum human interference, the machine can recognise patterns in data and use those patterns to make decisions. There are supervised and unsupervised machine learning algorithms. Supervised machine learning algorithms train machines by looking at examples. Machine learning algorithms that are not supervised, on the other hand, do not need labelled data, they operate independently to uncover previously unseen patterns and data.
Semi-supervised machine learning techniques combine supervised and unsupervised algorithms during training by integrating a small fraction of labelled data with a large percentage of unlabelled data. Regression and classification are the two basic tasks of supervised machine learning systems.

The algorithm's desired output in classification is a category value, whereas the expected result in regression is a real numeric value.

2.1.6 Feature Extraction from Natural Language

2.1.6.1 Natural language understanding

This is a field of study concerned with teaching computers to understand human language. This is normally done using a combination of techniques including syntactic, morphological and semantic analysis of a text.

2.1.6.2 Named-entity recognition

(Marshall, 2019) defines named-entity recognition as the selection and aggregation of key data such as entities discovered in text. It is also known as entity identification, entity extraction, or entity chunking. An entity is a word or a group of words that always refers to the same object. After an entity has been recognised, it is categorised into one of several categories, including persona names, places, organisations, amounts, percentages, and monetary values, among others. A named-entity identification machine learning model, for example, may recognise the phrase "Google LLC" in a sentence and classify it as a "business."

Both data labelling and the establishment of a training dataset are made feasible by defining items and categories. The training dataset can then be utilised to develop a text labelling algorithm.

2.1.7 Artificial Neural Networks

2.1.7.1 ANNs and Deep Neural Networks

Artificial neural networks (ANNs), are a type of algorithm that attempts to replicate how the human brain analyses and processes data (Frankenfield, J. 2020).

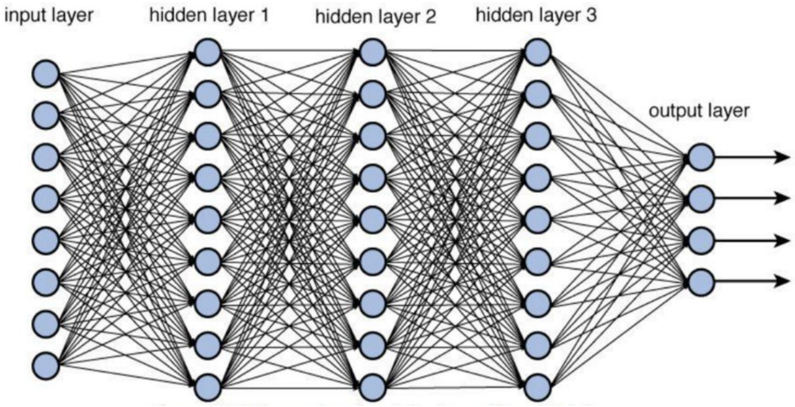


Figure 1. A Deep Neural Network

In classification, neural network techniques can be used to determine the target of an item. Classifiers vary in their ability to categorise items into multiple categories.

Deep neural networks (Deep Learning) address difficult data-driven problems by using artificial neural networks that replicate the behaviour of the human brain.

Deep learning is the name given to neural networks with multiple layers, often known as stacked neural networks.

2.1.8 WEKA, Waikato Environment for Knowledge Analysis

Waikato University in New Zealand developed and open-sourced WEKA (now at version 3.8.5) as a free machine learning solution. WEKA comes with useful utilities for data pre-processing and analysis, as well as well-tuned implementations of various algorithms ready to use.

2.1.9 Asset Valuation

The act of determining an asset's current value is known as valuation (Damodaran, 2012). The value is generally a positive real number that denotes a valuation in some legal tender, and the basic method for determining the approximate value is statistical regression.

2.1.9.1 Machine Learning for Car Valuation

To forecast the value of a variable, various regression procedures are utilised. Random forest regression, Lasso regression, Bagging regression, XGNoost, Multiple regression, K-Nearest-Neighbours regression, Adaboost regression and Ridge regression are some of the regression techniques that can be used.

Supervised machine learning methods such as decision trees, linear regression analysis, naive bayes and k-nearest neighbours were applied on vehicle data taken from newspapers to estimate the value of second-hand automobiles in Mauritius, according to (Pudaruth, 2014). To determine which algorithms, produce the best results, the forecasts were examined and contrasted.

2.1.9 Web Scraping

Web scraping, sometimes referred to as web harvesting, is a method of extracting data and material from a website utilizing software bots. The extracted information is gathered and exported in a usable manner.

Web scraping bots have to be programmed to recognise and extract data represented in a variety of mark-up formats such as JSON, XML, HTML and others.

Scraping, if done incorrectly, can result in intellectual property theft and the development of an inappropriate competitive advantage for businesses.

The legality of online scraping differs from country to country. This may be against certain websites' terms of service, but the terms enforcement is unclear.

The following are some of the difficulties that can arise when executing web scraping:

- Bot blocking measures: Administrators can use a variety of techniques to deter web scrapers. Examples include using dynamic pages built on AJAX, Captcha solutions, Visitor authentication and IP address blocking.
- Geo-blocking: This is the practice of restricting access to online data based on the users' location. This can be in the form of a white list or a black list.
- Complex web page structure: Because web pages vary, dissimilar scrapers are required for different target sites.
- Firewalls: new hardware and software technology makes it far easier to detect non-human web activity. If a website detects web scraping, the scrapper's Internet address can be blacklisted.
- Website structure evolution: To provide excellent services and improve user experience, websites frequently alter their content and user interface. This provides a challenge to scrapers because they must be appropriately updated to adapt to changes in the data representation of each site they crawl.

2.2 Empirical literature review

2.2.1 Raphael Kieti M.

Vehicle valuers in Kenya use experts' opinions to estimate a vehicle's worth. This method of valuing used cars has resulted in disparities in the values reported by different experts.

Kieti looked at the market, income, and price vehicle valuation methods, highlighting their advantages and disadvantages.

The algorithm chosen and implemented for forecasting the value of used cars was multiple regression analysis (MRA). MRA aided in the simulation of the relationship between a vehicle's worth and its determinants.

This algorithm illustrated how the value of a car was changed by changes in each of the factors (correlation).

The condition of a car, its mileage, type of vehicle, engine capacity, and colour were all crucial aspects (variables) that influenced its worth.

The MRA approach was able to ensure accurate vehicle valuation estimates (Kieti, 2005).

2.2.2 Hammad Hai & Haydn Ramanna Sonnad

Hai's (2019) study employed vehicle telemetry to create a vehicle valuation model. In his study, he used both vehicle and non-vehicle sources to obtain data. Telemetry refers to information obtained from a vehicle's sensors including its overall condition, its speed, year of fabrication, mileage, time spent in diverse weather conditions and the battery (status and charging capacity). To secure the vehicle data and avoid data manipulation from third parties, Hai attached a verified blockchain record to a vehicle to store the collected data.

A predictive valuation model was then created which removed the need for third parties' verification during a vehicle transaction between individuals. Real-time updates to the online model were used to keep the model updated.

2.2.3 Kaneeka Vidanage & Amjadh Ifthikar

According to (Ifthikar & Vidanage, 2018) due increase in the number of vehicles being purchased, the manual process of vehicle valuation was tedious, time consuming, and ineffective. The study aimed at solving the process of manual vehicle valuation by implementing a vehicle valuation system using web scraping and machine learning.
The proposed system has a web front for users to input data of the vehicle to be valued. An API (Application Programming Interface) was then used to cross-validate the data input with details of vehicle makes and models around the world. The details of the vehicle were sent to a web crawler which was programmed to collect enough data of vehicles identical to it. The web crawler scraps data from different websites that let the public advertise used vehicles by payment of a few. Websites that allowed people to advertise used vehicles at a free cost were not considered because the dealers may tend to quote unrealistic vehicle prices.

2.2.4 Zhang Yuquan & Chang Jiangxue

Second-hand vehicle evaluation research was done by Jiangxue and Yuquan (2018) focused on improved replacement cost methods. The new rate, the selling price and the purchase of the second-hand vehicle was able to be retrieved after an individual input the relevant basic information into the system and the system carrying out the calculations.

Price index and direct methods were used to calculate the replacement cost which was based on evaluations of the cost price of the original vehicle and its price level currently.

For imported cars or eliminated products that couldn’t obtain market prices, the price index method was used, whose formula is as follows:

Æ = (Ç÷Bÿ) × Bï

In which Æ represents the Replacement Cost

In which Ç represents the Original Cost

Bï represents the Price Index evaluated

Bÿ represents the Price Index when the vehicle is purchased

In the case of ownership transference, the Direct Method / Rest Algorithm was instead used to assess the vehicle.

Other factors warranting the use of the direct method included: an economic shift in the company/industry, current vehicle market price, taxes on vehicle usage, vehicle purchase surcharges and education fees.

To enhance authenticity in the pricing of the second-hand vehicle, the evaluation fees had to be readjusted depending on several factors, namely; the supply and demand in the market, frequent car maintenance, current competition in the market amidst other technical conditions. An analytic hierarchy process was used to determine the weights of the different factors involved as this enabled the construction of a judgement matrix since it provided expert scoring.

The focus of the study was to design a vehicle valuation system with its base on the improved replacement cost method. Albeit, improvement needs to be done since the system function of the model was not complete

.

2.2.5 Sandbhor & Chaphalkar

Using Artificial Intelligence to predict land prices, Sandbhor & Chaphalkar (2013), made use of fuzzy logic and such AI prediction models as Expert Systems and Artificial Neural Networks.

Land-price forecast involves the frequent use of artificial neural networks. Such a network is trained using a back-propagation algorithm, assisting the network through error correction and conducting weight adjustments as per the corrections.

Other techniques used for land value prediction involved natural language processing, the support vector machine and the hidden Markov model.

In a nutshell, the use of Artificial Neural Networks with back-propagation algorithms emerged to be more accurate during land price prediction compared to other algorithms

2.3 Opportunities for improvement

Discussed below are some integral pieces that need to be addressed after examining the empirical literature.

According to (Ifthikar & Vidanage, 2018), the automobile valuation process has to be enhanced further, such as by establishing a mobile application to make the process of doing car value easier. The program can be connected to an OBDII device and plugged into a vehicle's On-Board Diagnosis port to extract crucial information about the state of an engine.

Car evaluation based on the enhanced replacement technique was implemented in Jiangxue & Yuquan's 2018 research. Because of the short system design period, the designed system application was user-friendly, but the system function was not flawless, as it did not provide an exact prediction of vehicle evaluation. As a result, investing more time enhancing their design could have resulted in more consistent results.

According to Kieti's 2005 study, using questionnaires to determine the relationship between a car's value and its contributing elements can be enhanced by employing neural networks to accomplish the same task and construct a forecasting model.

From Sundari and Devi's research, there is a need to investigate different prediction algorithms that can produce reliable forecasts when applied to actual datasets.

2.4 Conceptual model

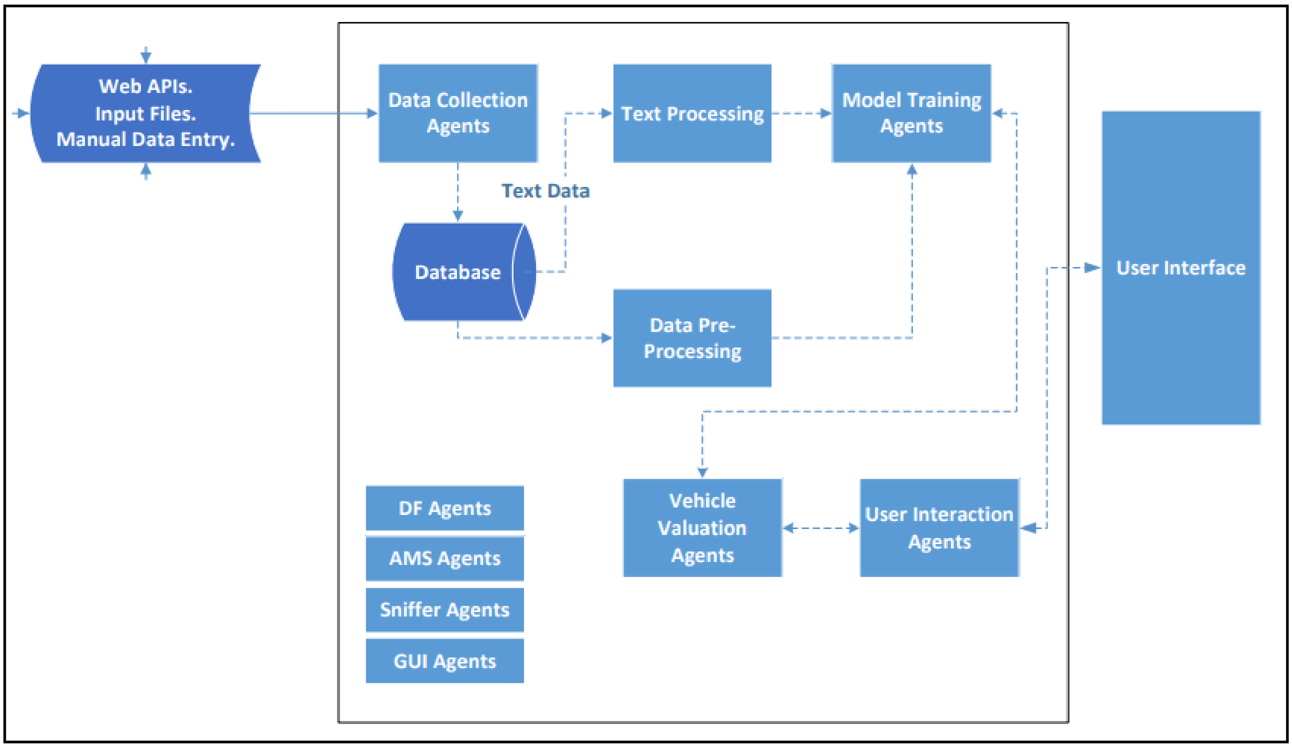


Figure 2. A high-level diagram of the conceptual design.

The proposed solution is abstracted in terms of agents co-existing and co-operating in a multi-agent platform to realize the goals of the study.

There are agents to carry out each important function of the research, starting with data collection, pre-processing, model fitting and instance prediction.

Natural Language understanding agents will perform instance identification and feature extraction from unstructured textual data.

User interaction agents will mediate between the user and the platform agents to relay input data and requests to the appropriate agents and display system output back to the user through the appropriate channels.

Additional agents such as Sniffers, Remote Monitoring Agent, Agent Monitoring System and the Directory Facilitator will be present in the final agent platform as they are part of the JADE framework.

2.5 Chapter Summary: Literature Review

Theoretical literature evaluation aided the researcher in better comprehending the several fields that this study encompasses. The results can be found in part 2.1 of the Literature Review under the heading Theoretical literature review.

The empirical evaluation of literature, on the other hand, was essential in developing a conceptual model of the answer that this study aims to provide.

After examining a lot of algorithms and processes applied in valuation determination, neural network-based regression and SVM based regression algorithms were favoured.

Chapter 3: Research Methodology

3.1 Introduction

This chapter of the report stipulates the tools, methods and processes that were employed to reach the identified research objectives.

Having selected a multi-agent architecture for the proposed design, it was necessary to select a multi-agent system design methodology to guide the design and implementation of the theorized solution. The Prometheus Design Methodology for Multi-Agent Systems was thus identified for this research.

3.2 Feasibility study

Considering that this research had placed upon it technical, financial and temporal constraints, as is the standard for this level of academic research, it was important to analyse various aspects of the research to establish whether it could be carried out within the set parameters.

3.2.1 Time feasibility

The allocated time for the project was found to be sufficient to carry out the different activities necessary for the successful completion of the study.

3.2.2 Technical feasibility

Having looked at the hardware and software resources available to the researcher, it was theorized that these were capable of handling the requirements of running the algorithms required for this research for moderate quantities of data.

3.2.3 Financial feasibility

This research would be carried out using open-sourced tools where possible, using equipment already available to the researcher, within the researcher's current location to minimized financial spending on this research.

3.2.4 Functional feasibility

The proposed solution through the fully functional prototype was expected to adequately solve the problems identified without necessitating radical changes in the problem environment.

3.3 Prometheus Design Methodology for Multi-Agent Systems

The mixed strategy prototype was designed and implemented following the guiding principles specified by the Prometheus design methodology.

Prometheus design methodology for multi-agent system design came into being as a culmination of pedagogical experience entwined with industrial application, as educators, scholars and industry specialists all participated in its formation.

The design methodology is different from any other multi-agent design methodologies as it explicitly enforces the procedural development of intelligent agents by stipulating functions, beliefs, agent plan descriptors, and events in an increasingly detailed manner.

The detailed process also encompasses the design of artefacts, and specific steps for realising the artefacts, and automation of verifying the design as well as the resulting artefacts.

In addition, the methodology provides a hierarchical structure that enables the design stage to occur with multiple, isolated levels of abstraction. This mechanism is vital for the simplification of the design and implementation of large multi-agent systems.

Prometheus methodology makes use of an iterative cycle of software development steps instead of a linear model that is common in other design philosophies. There is a consistent notation that serves to identify core agents, their life-cycle, capabilities, interactions, events, data and properties thereof.

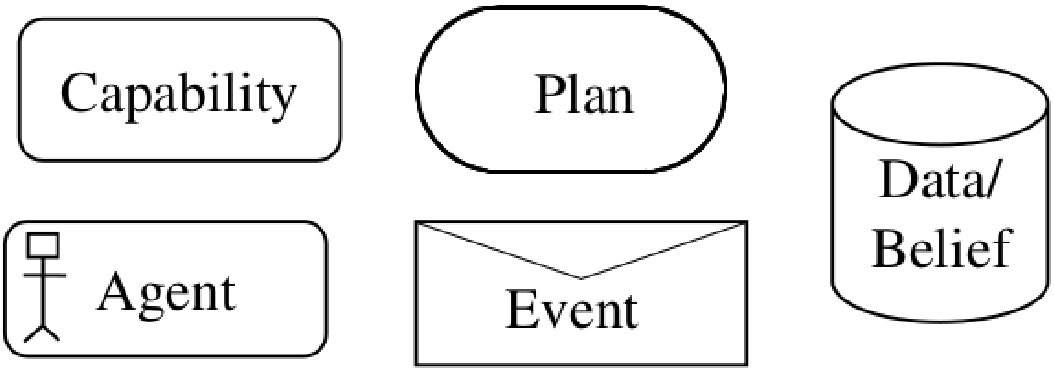


Figure 3. Common notation used for Prometheus design elements

3.3.1 Iterative Development Process

Prometheus multi-agent system design methodology decomposes the overall software design and development process into three main stages:

a. **System specification phase -** This stage identifies the core functions of the final multi-agent platform, with the expected inputs and outputs to be provided therefrom.
b. **Architectural design phase -** This stage focuses on the identification of the different agents that will live in the system and their interactions with other agents.
c. **Detailed design phase –** In this design stage, the internal workings of all the agents are specified to the finest details, along with the inputs and outputs of their functions.

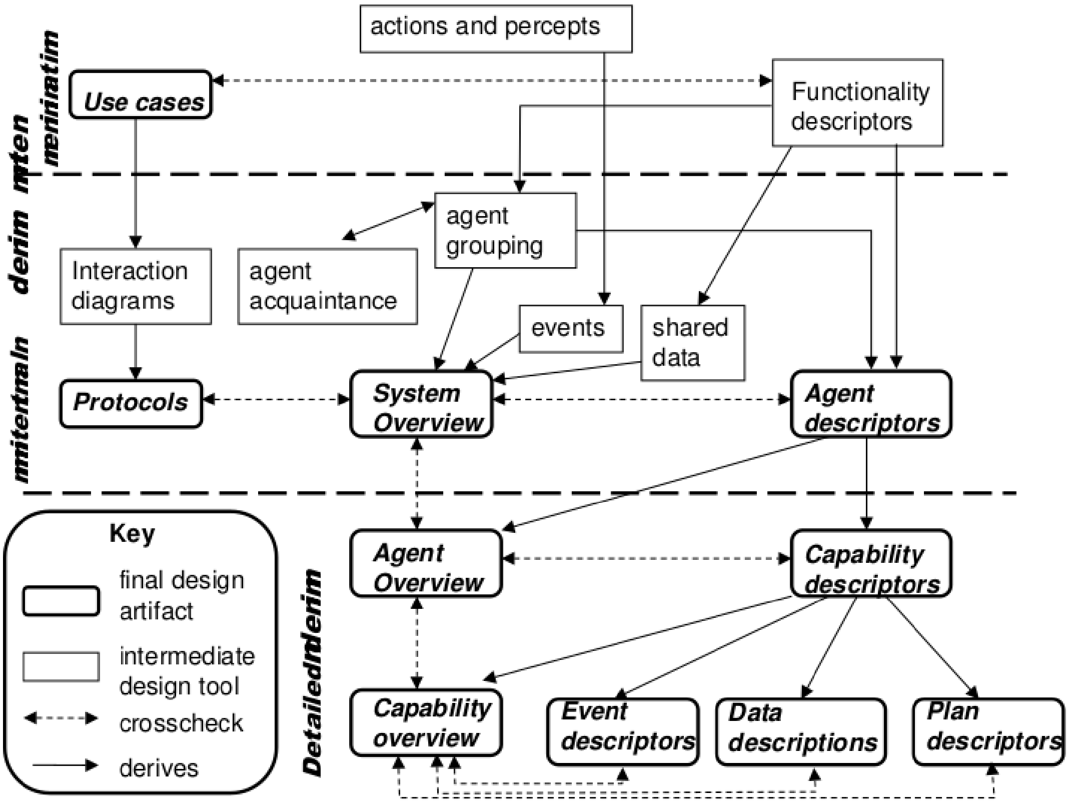


Figure 4. A stratified depiction of the 3 phases of Prometheus methodology

3.3.2 Phase 1: specification of the system

The central roles of the multi-agent prototype should be stated in this design stage. They include elements like inputs (percepts) coming from the environment through the input mechanisms into the system, and feedback arising from each agent's functions back to the environment, through the user-interface or output devices, which can be in form of actions or results of the action.

Inputs to the system, outputs from the system and their support mechanisms are the most important elements of the system specification phase.

For the software to be designed in this study, the following are the main percepts and actions expected.

Input/Percept	**Corresponding action**
Unstructured text	Feature extraction
Structured numeric data	Run algorithm to train model, predict a value
Category data	Convert to binary
Request for agent action	Service requested action

Table 1. Expected input data and corresponding actions.

These functionalities were also found to be necessary for the multi-agent platform

1. User-input prompting, reading and validation
2. Pre-processing both text and numeric data
3. Relaying system output back to the user
4. Initializing all system agents
5. Monitoring agent platform and taking corrective steps

3.3.3 Phase 2: Agent specification and architectural design

In this stage, all the core agents that will exist in the agent platform must be specified.
Building on the results of the previous design phase, extrapolating the core functions to create the required agent and system functionalities. Naturally, what follows is evaluating these roles concerning the principles of coupling and coherence, and determining the agents necessary according to the groupings of the said functionalities.
Agent descriptors are used in this stage to represent the functions of each agent.
An accepted method to gauge the simplicity and modularity of the design and coherence is to check if an agent name is simple and descriptive, conveying the agent functions without needing the addition of conjunctions such as and, commas, numbering and or.

To complete this phase, the following characteristics must be established about every single agent:

Number of each agent per platform.
The lifecycle of the agent.
When and how the agent comes to life.
When and how the agent stops.
Required resources.
The kind of data the agent handles and transmits.

3.3.3.1 Agent descriptor for the User interaction agent

Title	UserInteractionAgent
Roles	Interfaces with users; channels user inputs to appropriate agents and relays system prompts and feedback to the users
Cardinality	Single instance per platform
Start	On platform creation
Stop	On platform stop
Tasks	Updating user-interface, relay output to a user, read user-input, intra-platform communication

Percepts	Action events and data from and to a user interface, inter-agent communication entities
sociability	Input pre-processing agents, platform sniffers

Table 2. Sample agent-descriptor: User-interaction agent

3.2.3.2 Monitor and Restoration agent's descriptor

Title	MonitorAgent
Roles	Initializing and status monitoring of platform agents
Cardinality	Single instance per platform
Start	On platform creation
Stop	On platform stop
Tasks	Initializing agents, agent service querying, agent status monitoring, intra-platform communication, container mobility
Percepts	Instructions, status reports
sociability	Universal

Table 3. Sample agent-descriptor: System Runner

3.2.3.3 Descriptor for structured dataset pre-processing agents:

Title	Pre-processing Agent1
Roles	Cleaning structured data, category to binary conversion, missing value resolution
Cardinality	As many as necessary
Start	On platform creation, on-demand
Stop	On platform stop
Tasks	Pre-processing large datasets and single instances, intra-platform communication
Percepts	Datasets: training, evaluation, training and single instance for prediction

Sociability	User-Interface Agents, Instance-Prediction Agent, Sniffers

Table 4. Sample agent-descriptor: pre-processing agent

3.2.3.4 Descriptor for the text pre-processing agent:

Title	Text-Preprocessing-Agent
Roles	Feature extraction from unstructured text
Cardinality	As many as needed
Start	On platform creation
Stop	On platform stop
Tasks	Parts of speech tagging, Proper noun parsing from text (NER), identification of instances, intra-platform interaction
Data	Unstructured input text
Sociability	User-Interface Agent, Instance-Prediction Agents, Sniffers

Table 5. sample agent-descriptor: Text pre-processing agents

3.2.3.5 Agent descriptor for Model trainer agents

Title	ModelTrainingAgent
Roles	Models training, model testing
Cardinality	One per supported algorithm
Start	On platform creation
Stop	On platform stop
Tasks	Model training, testing and evaluation, intra-platform communication
Percepts	Datasets, Algorithm and hyperparameter specifications

Sociability	User-Interaction Agents, Instance-Prediction Agent, Sniffers

Table 6. sample agent-descriptors: Model training agents

3.2.3.6 Agent descriptor for Instance Prediction Agents

Title	VehicleValuation Agent
Roles	Instance value prediction
Cardinality	Single instance per platform
Start	On platform creation
Stop	On platform stop
Functionalities included	Loading pre-trained models, performing instance predictions, intra-platform communication
Percepts	Instance features and Algorithm description
Sociability	User-Interaction Agents, Pre-processor Agents, Interaction sniffers

Table 7. sample agent-descriptor: Instance valuation

Further agents shall be present on the platform, and this is because they are core agents in the JADE framework implementation of a multi-agent platform in Java. These agents are the Agent Management System agent, the Directory Facilitator agent, communication sniffers, and the Remote Monitoring Agent to provide a visual overview of the platform.

These agents are not included in architectural design because their functionality is not part of the interests and scope of this study.

3.2.4 Phase 3: The detailed design

Work done in this stage tackles the internal composition of each agent, and behaviours it will make use of to attain its objectives.

The different percepts, agent actions and their corresponding triggers and products are specified in detail in this part of the design methodology. The specifications must encompass the requisite input data, the applied logic or algorithm, the lifecycle of the event and the beneficiaries of the output.

As an illustration, a cyclic agent behaviour could listen for, receive, deserialize, filter and respond to message broadcasts, the undertake any actions required by the received communication. The example behaviour could, for instance, initiate another singleton behaviour to handle the action requested by the incoming message.

The output of this design state should be data descriptors, event descriptors, and detailed event descriptors.

Event descriptors are required to stipulate both sources and especially the significance of the said event along with the data needed and created by the event.

Plan descriptors, on the other hand, explicitly provide an identifier, plan steps, a description, the triggering event type, a context description including the plan lifecycle and requisite data types.

A data descriptor identifies functions and attributes of each class utilized to encapsulate data within the system.

A TemplateAgent class was created to contain functions and variables common to all agents, and each core system agent was a subclass of the TemplateAgent class.

3.2.4.1 Common Agent-Actions and Percepts

Interaction among agents on the JADE agent platform is in the form of serialized Percepts and AgentActions which must be deserializable for successful communication.

In the design artifacts from the detailed system specification phase, the implementation should permit percepts to encapsulate generic datasets, instances, descriptors and instructions, composed of native variables and custom classes. This will permit the researcher to save time on the communication ontology and use it on other parts of the prototype development.

3.2.4.2 Behaviours common to all agents

The agents implemented for the research prototype will implement the TemplateAgent class which will include inner classes implementing the following two kinds of agent behaviours:

- A one-shot behaviour – An abstract class that reacts to incoming communication. Each agent must provide its implementation to handle the different kinds of communication that is expected
- A cyclic behaviour – This agent behaviour will run throughout and listen to incoming communication. A custom implementation will specify the filters for the messages to accept and respond to.

3.2.4.3 Common functionality

All agents in the mixed strategy prototype must be able to perform every single one of the following functions:

- Searching for agent services on the Directory service
- Receiving both broadcast and specifically targeted messages from other agents on the same or different platforms.
- Upon creation, self-registering on the platform
- Self-deregistering upon demise.
- Confirming receipt of incoming communication

3.2.4.4 System Event Descriptors

a. *event descriptor: Performance evaluation*

Title	Performance Assessment Event
Activated by	User request, completed training
Lifetime	Model testing duration
Percepts	Test dataset
Actor	Prediction and training agents
Recipients	User-Interface agents, Prediction Agents

Table 8. event descriptor: Model evaluation

b. *GUI Action*

Title	User-Interface Event
Activated by	GUI interaction
lifetime	System processing duration. From user input to system output
Percepts	training dataset, vehicle instance features, system output, user action on the GUI
Actor	UI agents
Recipient	Preprocessing agents, Model trainer agents, Instance valuation agents

Table 9. event descriptor: User interface event

c. *Instance valuation event descriptor*

Event Name	Instance prediction event
Activated by	Instance features preprocessing, algorithm selection, user request
Lifetime	Order of microseconds. Prediction with a trained model takes negligible time
Percepts	Instance features
Actor	Instance valuation agent

Recipients	User Interaction agent

Table 10. event descriptor: Vehicle instance valuation

d. event descriptor: Raw Text Preprocessing

Title	Text parsing action
Activated by	User input and request
Lifetime	Proportional to amount of text
Percepts	Unstructured text data
Actor	Raw text pre-processor agents
Recipients	User Interaction agent

Table 11. event descriptor: NLP event

e. Model training Action

Title	Model fitting event
Activated by	Algorithm selection, preprocessed data and user request
Lifetime	Model training duration
Percepts	Training and evaluation datasets
Actor	Model fitting agents
Recipients	Instance prediction agents, User-interface agents

Table 12. event descriptor: Model training

3.4 Research procedure

This study followed the overall steps shown in the diagram below.

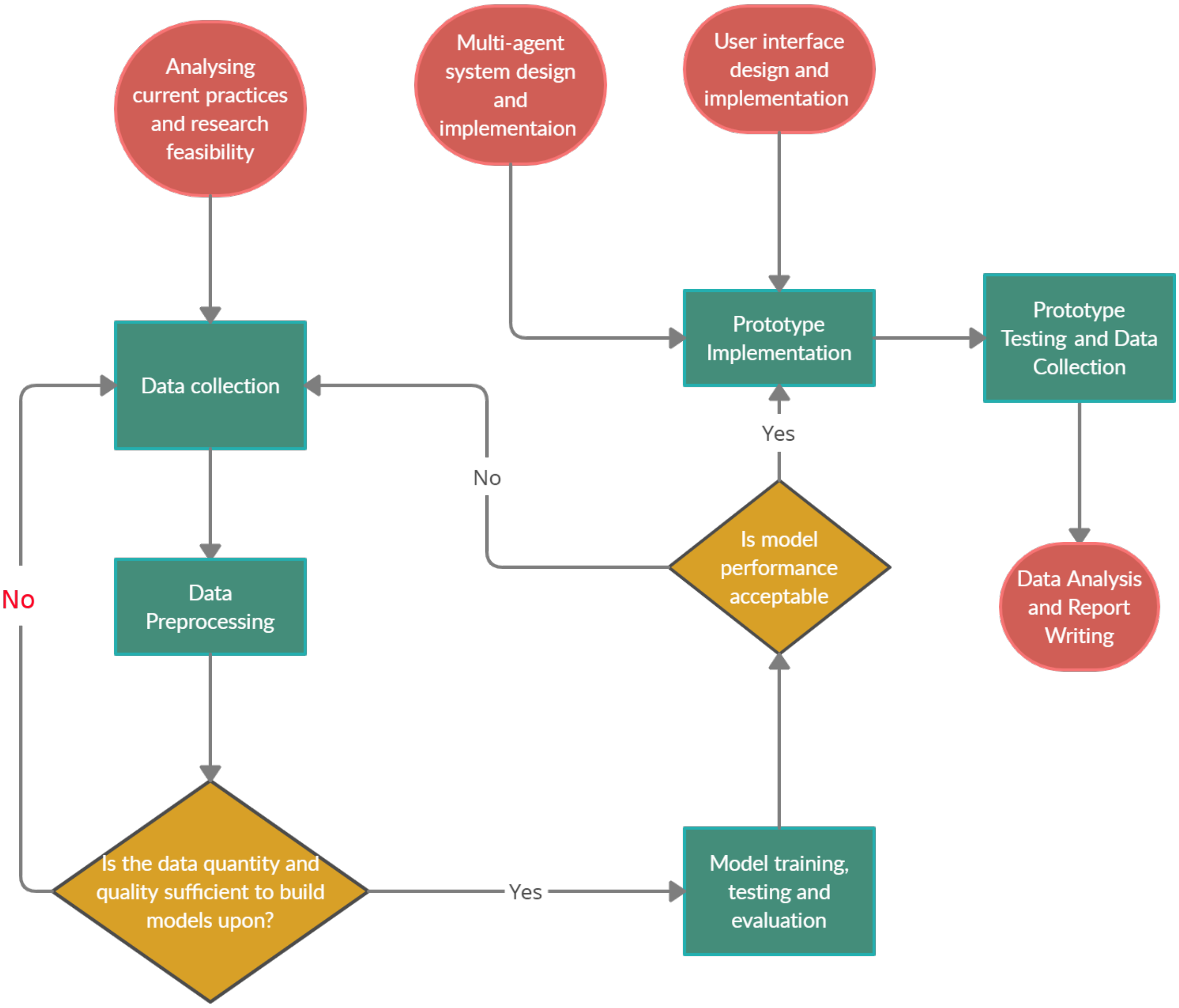

Analysing current practices and research feasibility
Multi-agent system design and implementaion
User interface design and implementation
Prototype Implementation
Prototype Testing and Data Collection
Data collection
No
Yes
Is model performance acceptable
Data Analysis and Report Writing
No
Data Preprocessing
Is the data quantity and quality sufficient to build models upon?
Yes
Model training, testing and evaluation

Figure 5. General flow of the research process

3.4.1 Resource aggregation stage

The requisite legal, technical, logistical, physical, financial, and human resources were identified and plans were put in place for their acquisition in advance so that the study would not be held back by a lack of the said resources.

This involved acquisition of data recording tools, a large capacity server to use a data repository for collected data, as well as permits to access and utilize third-party data.

3.4.2 Methods Used to Collect Data

- Scraping data from online car dealer websites. The researcher made use of web crawlers to amass data from twenty-two online sources. These crawling bots were designed and executed following the criteria for fairness expressed in the following segment.
- Pre-collected data. The researcher was able to retrieve both current and historical vehicle pricing information from a couple of vehicle dealerships. These datasets were shared in the

form of excel files.

- Experience. To understand current practices in used vehicle valuation, the researcher presented them as a client in several insurance firms, car resellers, brokers and valuers. The information was collected by asking explicitly how they did their valuation, and also presenting resolution cases.
- Email surveys and questionnaires. The researcher sent emails to insurance brokers and valuers asking about their vehicle valuation process.

3.4.2.1 Considerations for web scraping

Web scraping can easily turn into a denial-of-service attack on the target web server if not done carefully. As such the following considerations were used while designing the web scraping bots for this study:

1. For every request sent to a server, the bot would sleep for a proportional length of time before sending the next request.
2. Each queried resource was persisted in an RDBMS with the URL as a unique key to avoid making duplicate requests for the same content in future.
3. In addition, an informative user-agent header was sent with every request. The purpose for this was to make the intentions of the data collection clear and allow the affected system administrators a channel to contact the researcher with any concerns in that case that any materialized.
4. An unintended DoS attack is more likely when legitimate traffic is combined with crawler traffic. To mitigate this, the scrapers were executed during late hours of the night when web traffic to the servers of interest was at a minimum.
5. This study is not collecting data from car dealers to provide a competing service.

The lack of uniformity in the data was a drawback of getting the data from a variety of secondary sources. At random, some dealers left out information like pricing, mileage, transmission type and colour amongst others.

3.4.3 Data Pre-processing

The gathered data was analysed and prepared for input into the chosen algorithms in this step of the study.

Outlier management, attribute selection, numeric value normalization, missing features, and one-hot encoding of nominal data including car manufacturer, models, gasoline types, and gearboxes were all part of the process.

The bulk of the pre-processing was done in Java-based Weka v3.8.5, and Python utilizing two open-source data science tools pandas and NumPy.

3.5 Unprocessed data repository

The researcher created a repository based on RDBMS to store raw data before it could be further processed. Using a relational database was a good choice for data collected by crawling the web, as the researcher could use database indexing to limit the number of HTTP requests sent to target web servers by first querying the stored data to see if the URL had already been fetched before.

To handle all XML, HTML and JSON responses, the web scrapers used an XML node traversal and manipulation library (cheerio) or the native support for JSON objects in Javascript and NodeJS to handle HTTP response bodies appropriately.

MySQL spin-off project MariaDB was selected to provide the RDBMS for this study. To connect to the database and conduct CRUD operations, the NodeJS scrapers made use of an ORM abstraction layer (sequelize library).

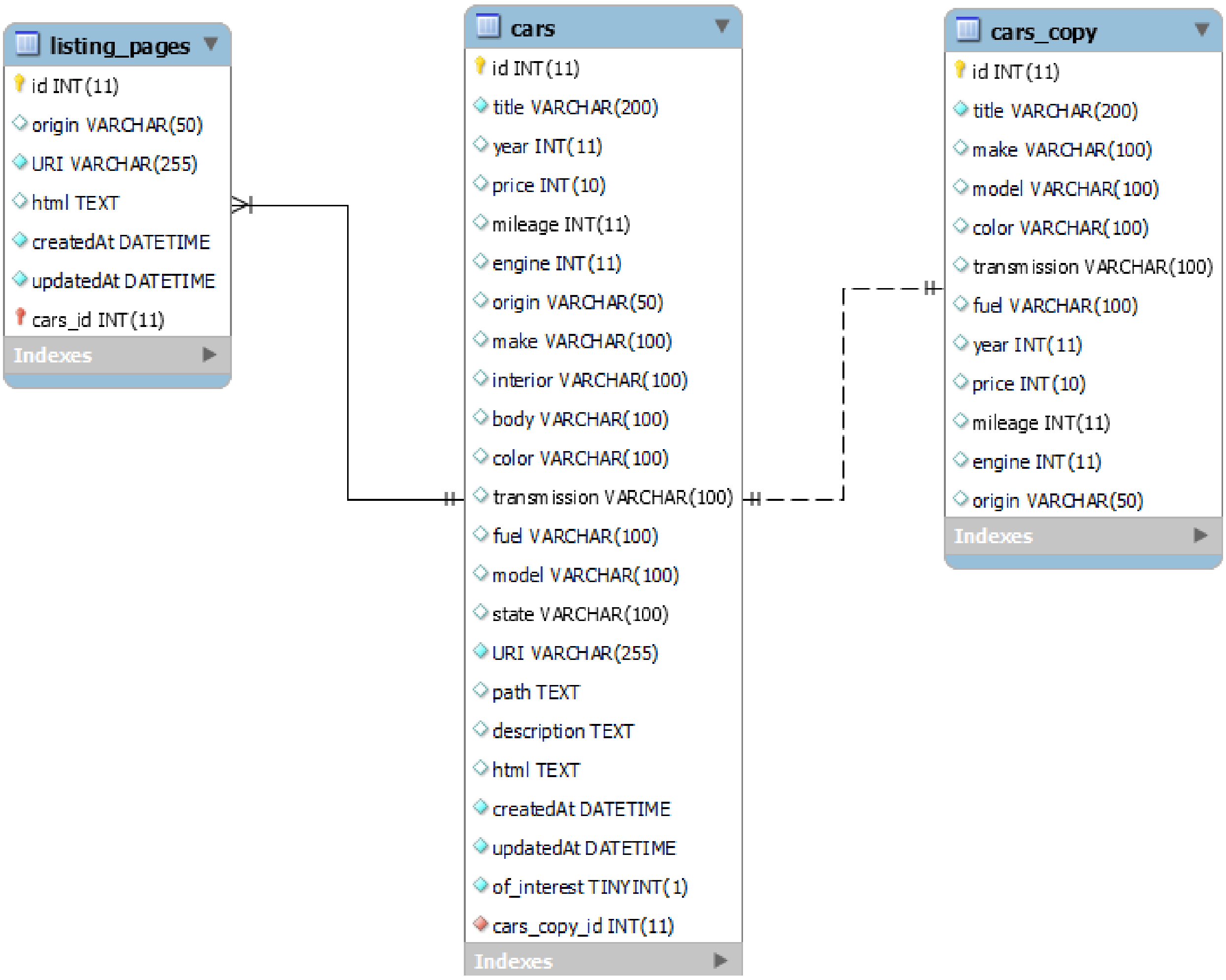


Figure 6. design of the repository database

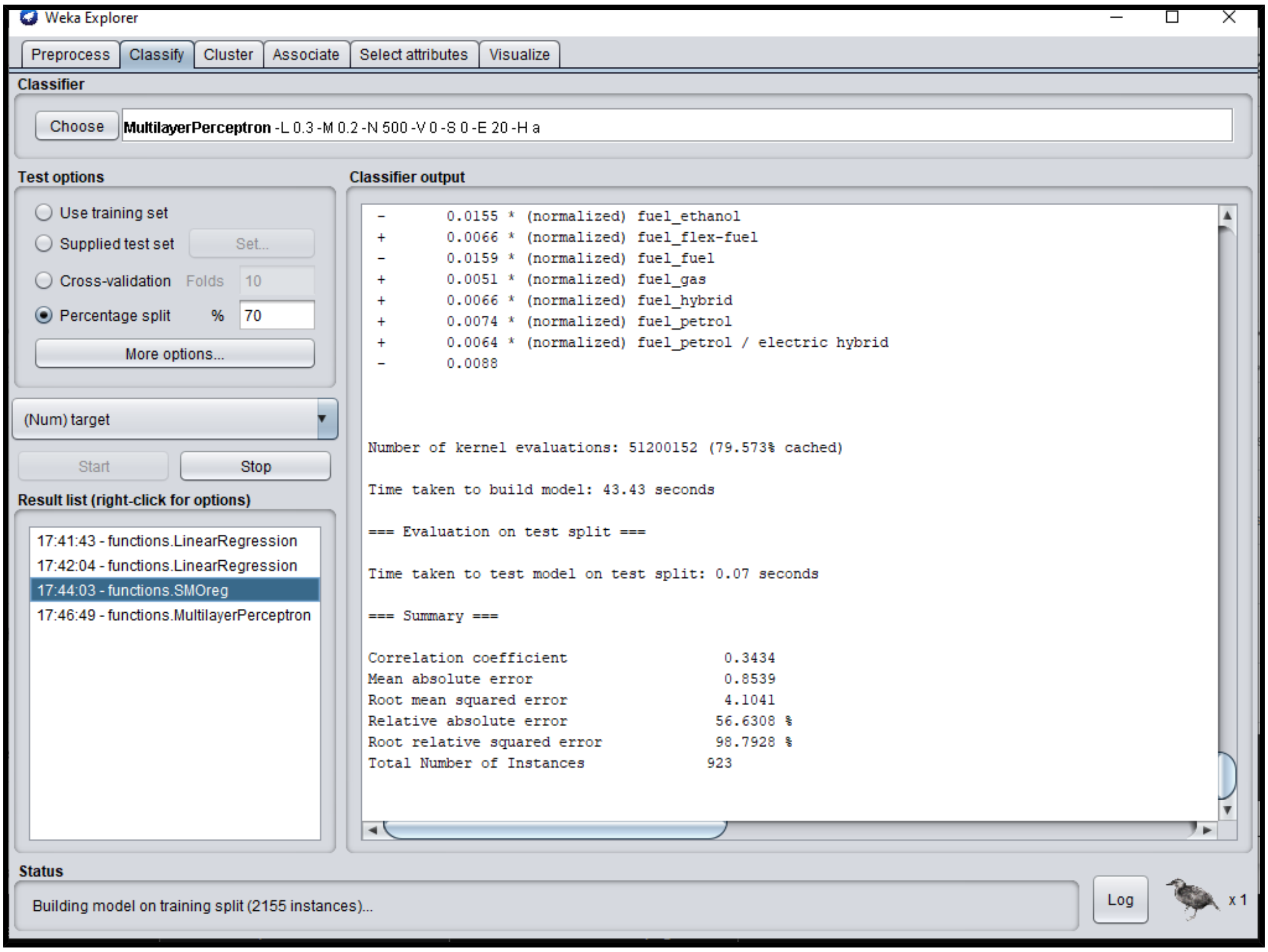
3.6 Algorithm training and comparison

From the various options available in WEKA, three regression algorithms were selected, namely Linear regression, Artificial Neural Network (MLP) and SVM regression (SMOreg).

For each selected algorithm, the model was trained on a dataset identical to the other two, and its performance metrics were obtained from running on an identical testing set and compared.

WEKA workbench allowed the research to compare the algorithm performance without having to fabricate its implementations of the chosen algorithms.

As can be seen in the screenshot below, on the WEKA workbench, multiple models can be trained and their performance is displayed on the interface for easy comparison.



MultilayerPerceptron -L 0.3 -M 0.2 -N 500 -V 0 -S 0 -E 20 -H a

```
 -       0.0155 * (normalized) fuel_ethanol
 +       0.0066 * (normalized) fuel_flex-fuel
 -       0.0159 * (normalized) fuel_fuel
 +       0.0051 * (normalized) fuel_gas
 +       0.0066 * (normalized) fuel_hybrid
 +       0.0074 * (normalized) fuel_petrol
 +       0.0064 * (normalized) fuel_petrol / electric hybrid
 -       0.0088



Number of kernel evaluations: 51200152 (79.573% cached)

Time taken to build model: 43.43 seconds

=== Evaluation on test split ===

Time taken to test model on test split: 0.07 seconds

=== Summary ===

Correlation coefficient                  0.3434
Mean absolute error                      0.8539
Root mean squared error                  4.1041
Relative absolute error                 56.6308 %
Root relative squared error             98.7928 %
Total Number of Instances              923
```

```
17:41:43 - functions.LinearRegression
17:42:04 - functions.LinearRegression
17:44:03 - functions.SMOreg
17:46:49 - functions.MultilayerPerceptron
```

Building model on training split (2155 instances)...

Figure 7. screenshot of algorithm parameters on WEKA workbench

Further data analysis was carried out and reported in the section 4.2

Chapter 4: Data Analysis, Prototype Design and Implementation

4.1 Overview

Raw data analysis, attribute selection, data pre-processing, in-depth data analysis and use of the resulting dataset to implement working models are all covered in this chapter.

4.2 Current vehicle valuation practice analysis

The study's initial practical step was to gain a thorough understanding of existing procedures and regulations of the data at hand.

4.2.1 Contemporary used car valuation methods

The researcher sent out 23 requests for valuation practices to vehicle valuers, dealers and insurance company agency brokers. From this targeted group, 2 valuable responses were received.
Some of the others requested in-person meetings that never came to fruition, while others cited restrictions imposed by a non-disclosure agreement from sharing this information.

The survey had only one question, which asked the respondent to explain how they work, to outline their processes for automobile evaluation including what variables are vital to the process. The intended use for this information was also made clear to alleviate the fear of empowering a competitor.

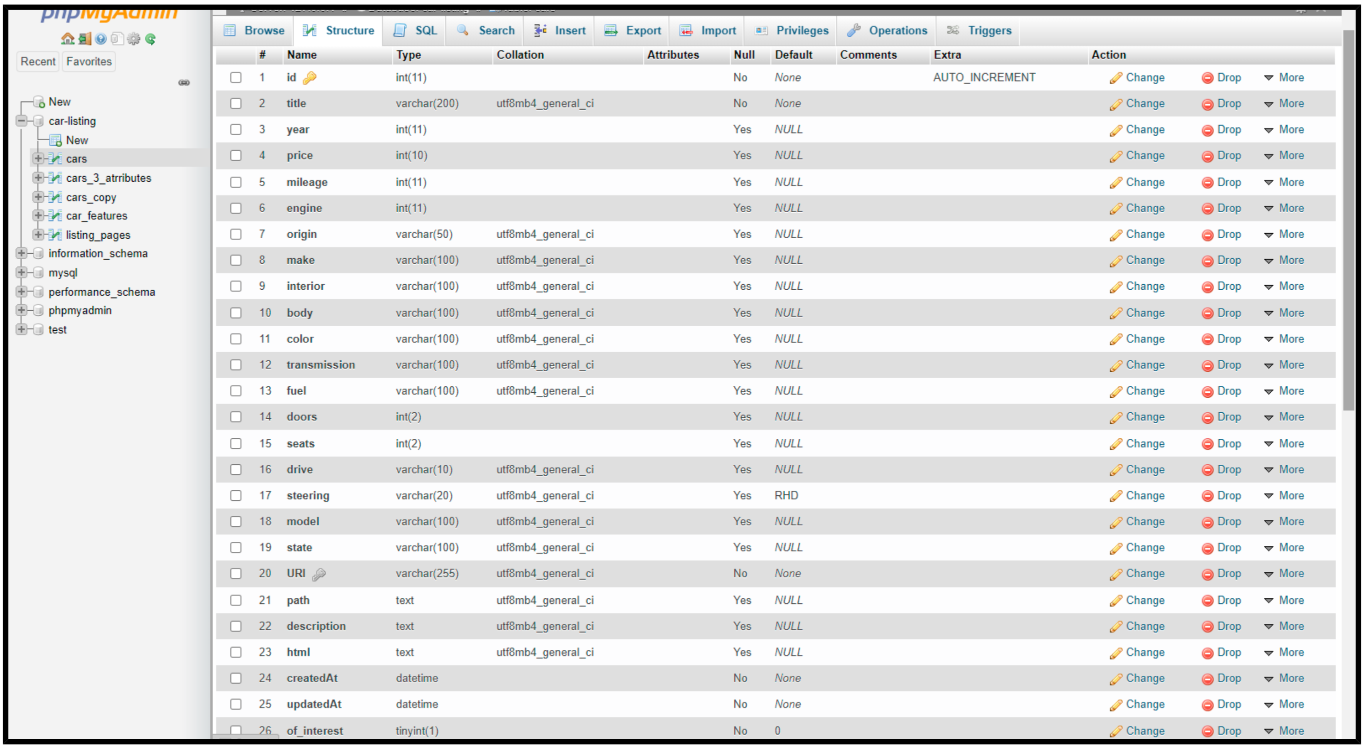
Both of the respondents' approaches emphasized a dependence on mathematical formulas that focus on yearly compounded devaluation calculations and deducting it from the car's original purchase price taking inflation into account.

4.2.2 Inspection of unprocessed data repository

As stipulated in the scope of this study, the focus is on car valuation in Kenya, and thus all of the information in this study comes from records gathered from several car dealerships in Kenya.

Approximately 350,000 rows of data were collected using the strategies outlined in Data Collection Methods. This figure dropped to approximately 250,000 upon clearing out the irrelevant rows which were mostly advertisements for spare parts and vehicle leasing services. The knowledge acquired from analysing the unfeasible data aided in the data collection process, tightening the collection and filtering criteria to reduce the amount of garbage ending up in the repository.

Important attributes in the resulting dataset were: model, body, title, year, engine capacity, price, transmission, mileage, make, number of doors, description, colour, fuel type, number of seats and interior (type and colour).



	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action		
☐	1	id	int(11)			No	None		AUTO_INCREMENT	Change	Drop	More
☐	2	title	varchar(200)	utf8mb4_general_ci		No	None			Change	Drop	More
☐	3	year	int(11)			Yes	NULL			Change	Drop	More
☐	4	price	int(10)			Yes	NULL			Change	Drop	More
☐	5	mileage	int(11)			Yes	NULL			Change	Drop	More
☐	6	engine	int(11)			Yes	NULL			Change	Drop	More
☐	7	origin	varchar(50)	utf8mb4_general_ci		Yes	NULL			Change	Drop	More
☐	8	make	varchar(100)	utf8mb4_general_ci		Yes	NULL			Change	Drop	More
☐	9	interior	varchar(100)	utf8mb4_general_ci		Yes	NULL			Change	Drop	More
☐	10	body	varchar(100)	utf8mb4_general_ci		Yes	NULL			Change	Drop	More
☐	11	color	varchar(100)	utf8mb4_general_ci		Yes	NULL			Change	Drop	More
☐	12	transmission	varchar(100)	utf8mb4_general_ci		Yes	NULL			Change	Drop	More
☐	13	fuel	varchar(100)	utf8mb4_general_ci		Yes	NULL			Change	Drop	More
☐	14	doors	int(2)			Yes	NULL			Change	Drop	More
☐	15	seats	int(2)			Yes	NULL			Change	Drop	More
☐	16	drive	varchar(10)	utf8mb4_general_ci		Yes	NULL			Change	Drop	More
☐	17	steering	varchar(20)	utf8mb4_general_ci		Yes	RHD			Change	Drop	More
☐	18	model	varchar(100)	utf8mb4_general_ci		Yes	NULL			Change	Drop	More
☐	19	state	varchar(100)	utf8mb4_general_ci		Yes	NULL			Change	Drop	More
☐	20	URI	varchar(255)	utf8mb4_general_ci		No	None			Change	Drop	More
☐	21	path	text	utf8mb4_general_ci		Yes	NULL			Change	Drop	More
☐	22	description	text	utf8mb4_general_ci		Yes	NULL			Change	Drop	More
☐	23	html	text	utf8mb4_general_ci		Yes	NULL			Change	Drop	More
☐	24	createdAt	datetime			No	None			Change	Drop	More
☐	25	updatedAt	datetime			No	None			Change	Drop	More
☐	26	of_interest	tinyint(1)			No	0			Change	Drop	More

Figure 8. PhPMyadmin Visualization of Raw Data Table

Omitted feature costs were some of the initial checks conducted on the data. Rows that excluded critical values such as the model, year of manufacture and price of the cars were relinquished instantly.

There were 195,435 records in total in the final dataset, with the following defects: 31,091 records were without a value for mileage

- 90,675 colour description records were omitted
- 7,633 records lacked transmission descriptors
- 78,612 accounts lacked fuel type cost
- 114,594 did not address engine capacity
- 1900 entries were under-priced as they advertised car hire services

Only 17,867 records had valid values for all the attributes.

By using a look-up table, some values for the missing attributes could be recovered, bringing the number of rows without missing attributes to 54,000.

Sample data from the look-up table;

make: nissan
model: note
year: 2017

seats: "{\"5\":39}"
doors: "{\"5\":40}"
engine: "{\"1100\":1,\"1200\":41,\"1600\":2}"
fuel: "{\"hybrid(petrol)\":17,\"petrol\":52,\"hybrid\":2}”
transmission: "{\"automatic\":91,\"manual\":7,\"cvt\":4}"
drive: "{\"4wd\":2,\"2wd\":3}"
body: "{\"hatchback\":58}"
color: "{\"blue\":1,\"pearl\":10,\"red\":2,\"light_blue\":1,\"white\":14}”

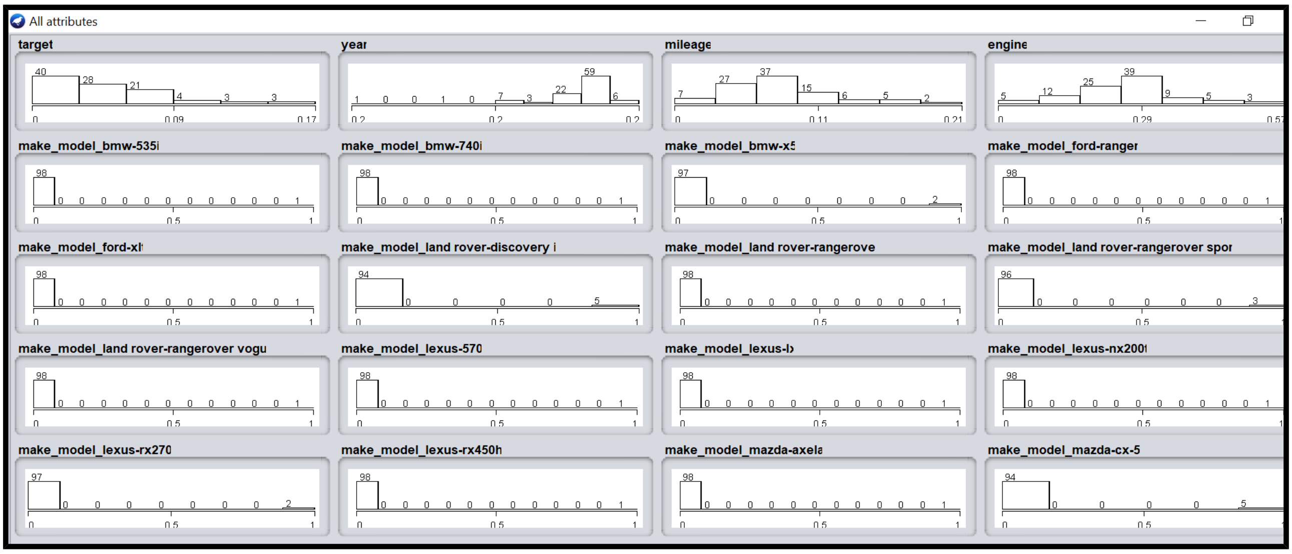


Figure 9. WEKA pre-processor visualization window for all attributes

4.3 Selection of attributes

After highlighting the data flaws, it became clear that it be impossible to train and test the prediction model using all attributes. Some features had to be excluded based on certain criteria.

Given that one of the goals of this research was to enhance the evaluation of automobiles that appeared in unstructured text, it was critical to examine what information is typically supplied to characterise cars in English writing. Searching news archives for vehicles, revealed a high recurrence of cars classified by make, colour, model, body style, and year in decreasing order. Making colour essential would require the removal of over half of the data set, while automobile values of homogenous make and model varied so widely across a period of time that the year of manufacture could not be ignored if the model was to function accurately.

It was established that there would be a minimum of two heterogeneous sets of prediction models; one that worked on year, manufacturer and model to perform prediction of cars found in the text, and a second to work on a collection of other characteristics that would be chosen.

On performing Principal Component Analysis in WEKA, the correlation of the attributes to the target value were evaluated with every attribute that scored less than 70% of the highest-ranking being dropped from the research.

That left only 7 attributes in the dataset; manufacturer, model, fuel type, transmission, year of manufacture, mileage and engine capacity in addition to the price of the vehicle.

Features selected for training model for a textual dataset

Attribute	**Sample value**
Manufacturer	Honda
Type	CRV
Year of Manufacture	2013
Value (KES)	1,450,000

Table 13. Sample features for Text based model

Attributes selected for general prediction

Attribute	**Sample values**
Manufacturer	Honda
Type	CRV
Fuel type	Petrol
Drivetrain	Automatic
Distance covered (km)	100000
Year of manufacture	2013
Size of the engine (cc)	2400
Cost (KES)	1,450,000

Table 14. Sample features used in main model

Discarded features

Attribute	**Sample value**
Number of doors	5
Airbags	2
Number of seats	7

Body	SUV
Interior	Leather
Exterior Colour	Red
Drive	4WD
Interior colour	Grey

Table 15. The ignored attributes

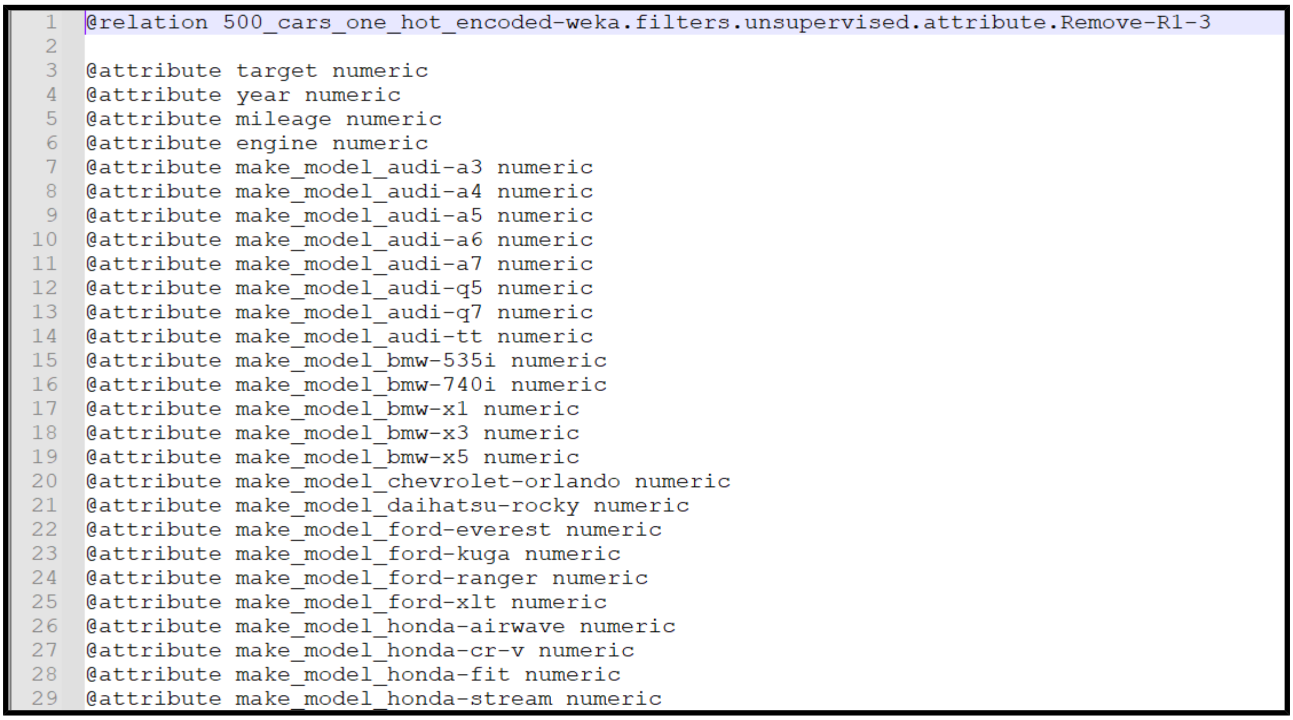
4.4 Data pre-processing

As the raw data was in an RDBMS repository, SQL queries were needed to filter and export the data identified for the study to a CSV format which could easily be ingested directly into most algorithm implementations in the form of a matrix.

Sample queries are included in the appendix.

Raw data was duplicated to additional schemas to enable rollbacks in case the evolution of the research necessitate it.

After fitting nominal to binary to categorical data with the pandas library for data science's one-hot encoding functionality, all numerical values were converted into a sparse matrix of binary values.



```
@relation 500_cars_one_hot_encoded-weka.filters.unsupervised.attribute.Remove-R1-3

@attribute target numeric
@attribute year numeric
@attribute mileage numeric
@attribute engine numeric
@attribute make_model_audi-a3 numeric
@attribute make_model_audi-a4 numeric
@attribute make_model_audi-a5 numeric
@attribute make_model_audi-a6 numeric
@attribute make_model_audi-a7 numeric
@attribute make_model_audi-q5 numeric
@attribute make_model_audi-q7 numeric
@attribute make_model_audi-tt numeric
@attribute make_model_bmw-535i numeric
@attribute make_model_bmw-740i numeric
@attribute make_model_bmw-x1 numeric
@attribute make_model_bmw-x3 numeric
@attribute make_model_bmw-x5 numeric
@attribute make_model_chevrolet-orlando numeric
@attribute make_model_daihatsu-rocky numeric
@attribute make_model_ford-everest numeric
@attribute make_model_ford-kuga numeric
@attribute make_model_ford-ranger numeric
@attribute make_model_ford-xlt numeric
@attribute make_model_honda-airwave numeric
@attribute make_model_honda-cr-v numeric
@attribute make_model_honda-fit numeric
@attribute make_model_honda-stream numeric
```

Figure 10. Attribute-Relation File Format supported by W.E.K..A.

4.5 Splitting the dataset

The entire set was divided into an 80/10/10 training/evaluation/testing split with a random selection of rows in each split. Weka provides mechanisms for performing this split on the same dataset programmatically, allowing one dataset to be provided to the application instead of 3 separate

datasets.
This allows the three datasets to align after nominal to binary conversion, and dimensionality reduction by principal component analysis.

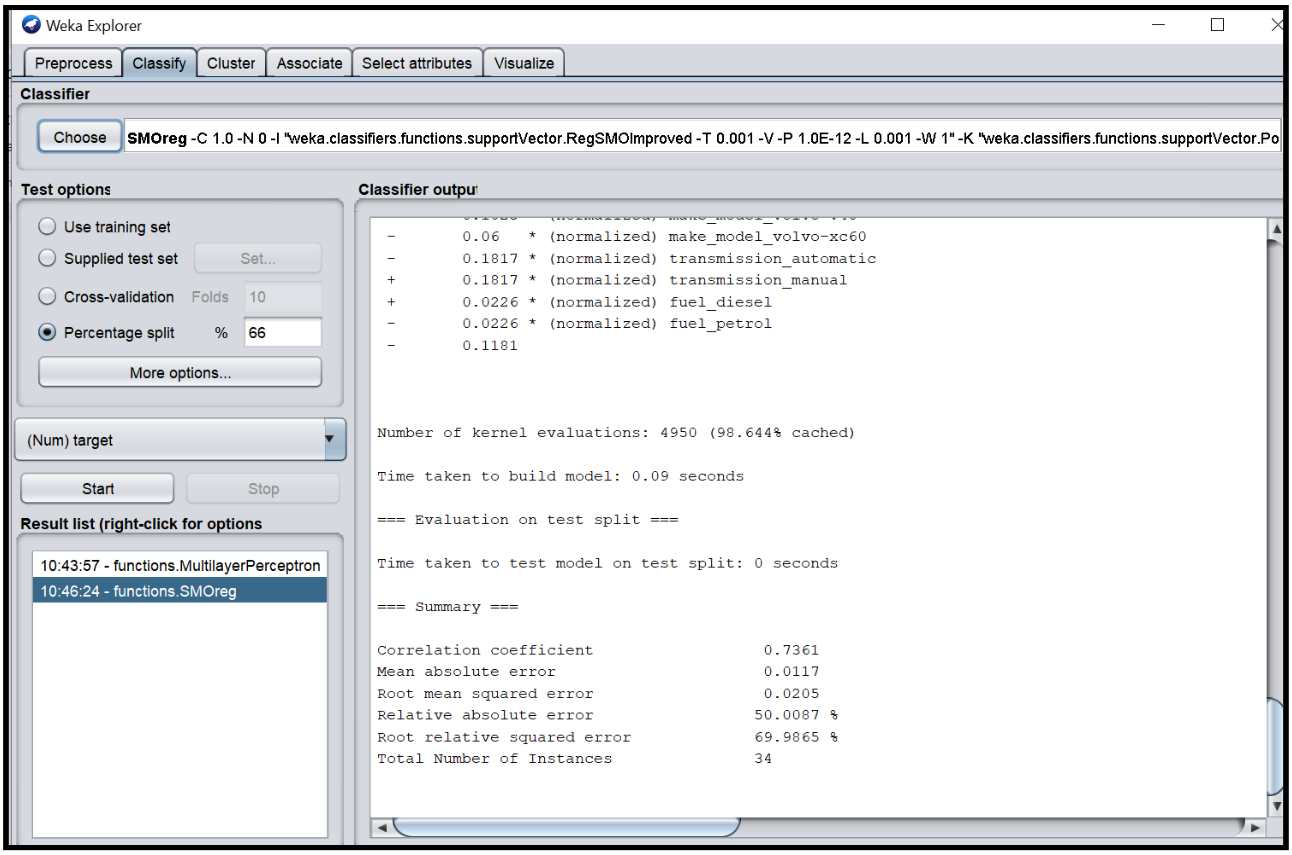
For test runs on the WEKA workbench, a 70/30 training/testing split was configured.

4.6 Algorithm testing and selection

It was vital to examine the performance of all available possibilities before deciding which algorithms the prototype should employ, and the WEKA workbench provided the ideal instrument for this task.

All obtainable regression algorithms were taught on a limited database of 500 records of data.

ANN (MLP) and SMOreg gave the most promising performance and were thus selected for future application in the study.



SMOreg -C 1.0 -N 0 -I "weka.classifiers.functions.supportVector.RegSMOImproved -T 0.001 -V -P 1.0E-12 -L 0.001 -W 1" -K "weka.classifiers.functions.supportVector.Po

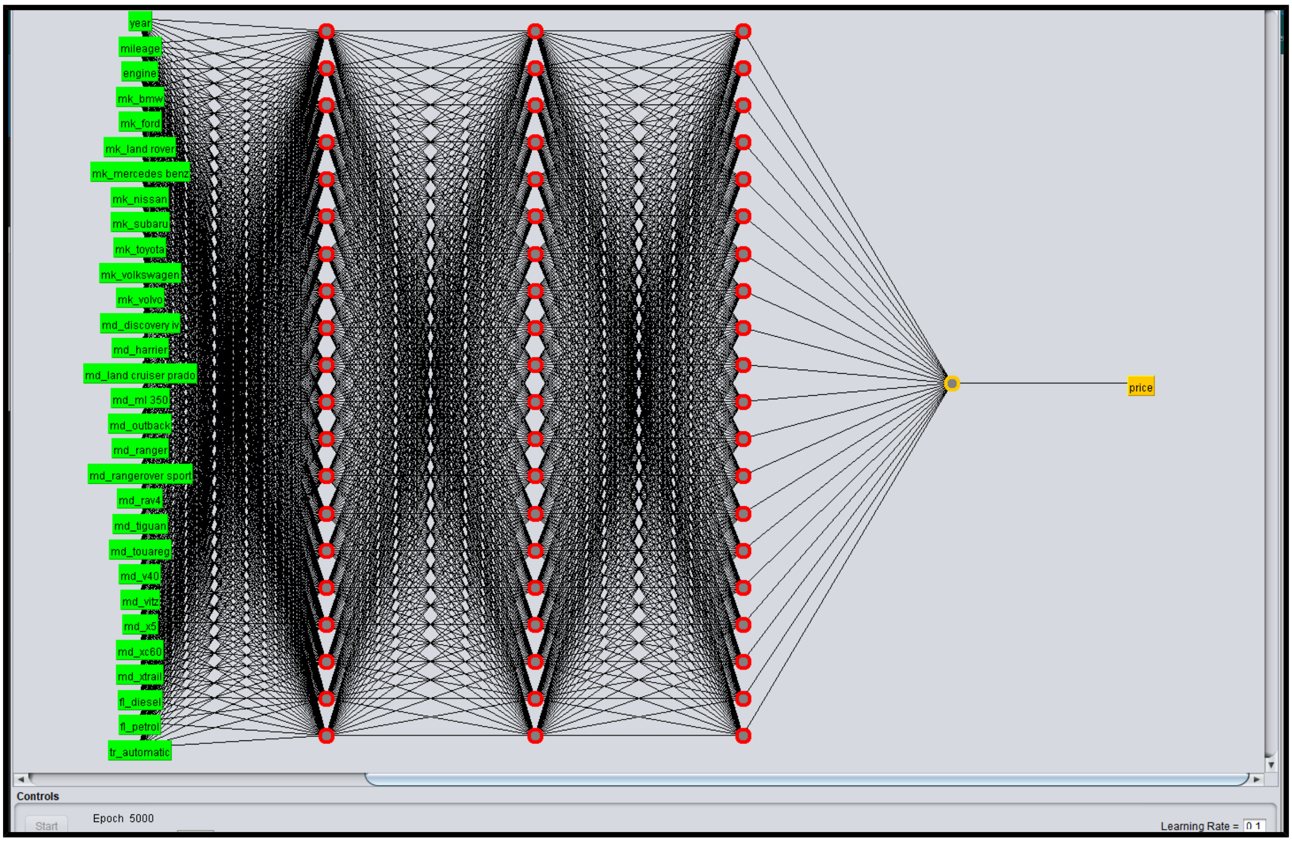
○ Use training set
○ Supplied test set
○ Cross-validation Folds 10
● Percentage split % 66

(Num) target

10:43:57 - functions.MultilayerPerceptron
10:46:24 - functions.SMOreg

```
 -       0.06   * (normalized) make_model_volvo-xc60
 -       0.1817 * (normalized) transmission_automatic
 +       0.1817 * (normalized) transmission_manual
 +       0.0226 * (normalized) fuel_diesel
 -       0.0226 * (normalized) fuel_petrol
 -       0.1181



Number of kernel evaluations: 4950 (98.644% cached)

Time taken to build model: 0.09 seconds

=== Evaluation on test split ===

Time taken to test model on test split: 0 seconds

=== Summary ===

Correlation coefficient                  0.7361
Mean absolute error                      0.0117
Root mean squared error                  0.0205
Relative absolute error                 50.0087 %
Root relative squared error             69.9865 %
Total Number of Instances               34
```

Figure 11. SMOreg algorithm performance after preliminary testing

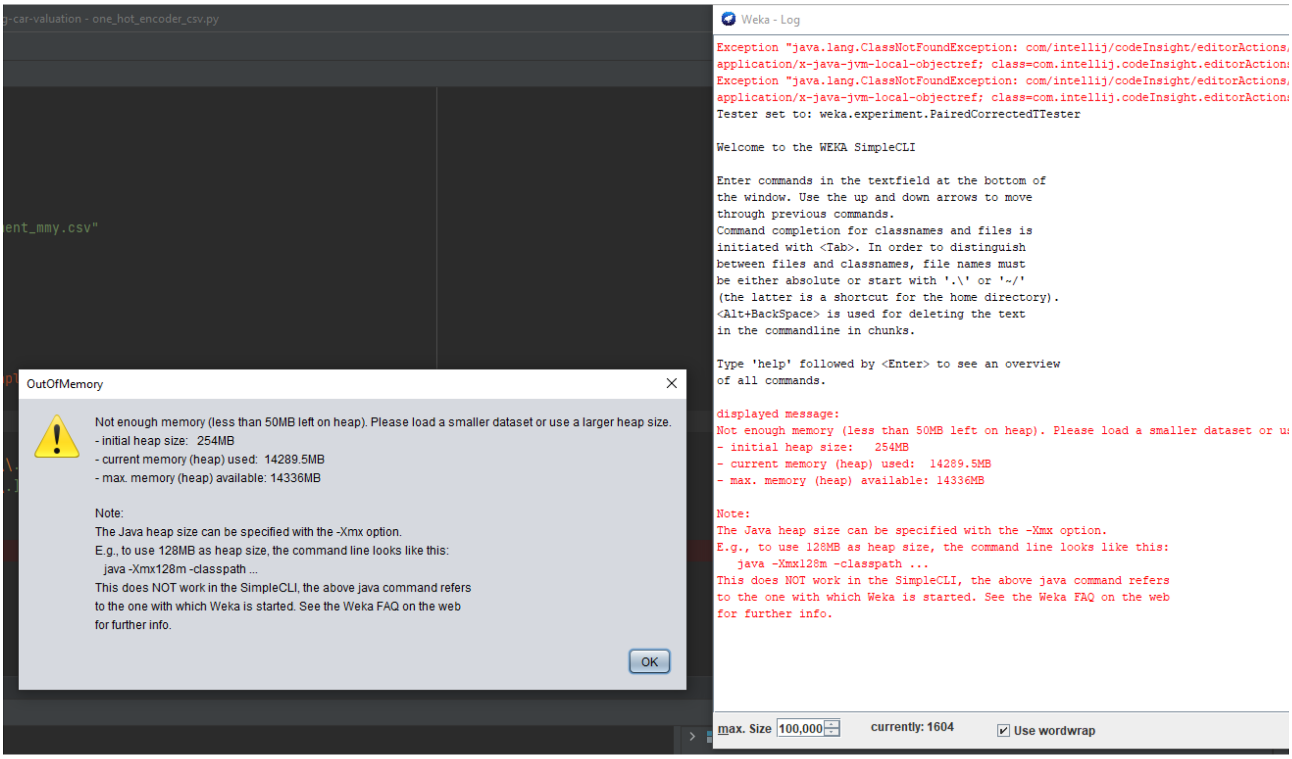


Epoch 5000

Learning Rate = 0.1

Figure 12. Visualization of ANN layers during preliminary testing

Performance and convergence issues necessitated a look into further options, and WEKA’s deep learning implementation WekaDeeplearning4j provided the Dl4jMlpClassifier which can be used to perform regression on GPU as opposed to the SMOreg and Multilayer Perceptron algorithms which could only run on a CPU.



```
Not enough memory (less than 50MB left on heap). Please load a smaller dataset or use a larger heap size.
- initial heap size:   254MB
- current memory (heap) used:  14289.5MB
- max. memory (heap) available: 14336MB

Note:
The Java heap size can be specified with the -Xmx option.
E.g., to use 128MB as heap size, the command line looks like this:
  java -Xmx128m -classpath ...
This does NOT work in the SimpleCLI, the above java command refers
to the one with which Weka is started. See the Weka FAQ on the web
for further info.
```

```
Exception "java.lang.ClassNotFoundException: com/intellij/codeInsight/editorActions,
application/x-java-jvm-local-objectref; class=com.intellij.codeInsight.editorAction:
Exception "java.lang.ClassNotFoundException: com/intellij/codeInsight/editorActions,
application/x-java-jvm-local-objectref; class=com.intellij.codeInsight.editorAction:
Tester set to: weka.experiment.PairedCorrectedTTester

Welcome to the WEKA SimpleCLI

Enter commands in the textfield at the bottom of
the window. Use the up and down arrows to move
through previous commands.
Command completion for classnames and files is
initiated with <Tab>. In order to distinguish
between files and classnames, file names must
be either absolute or start with '.\' or '~/'
(the latter is a shortcut for the home directory).
<Alt+BackSpace> is used for deleting the text
in the commandline in chunks.

Type 'help' followed by <Enter> to see an overview
of all commands.

displayed message:
Not enough memory (less than 50MB left on heap). Please load a smaller dataset or u:
- initial heap size:   254MB
- current memory (heap) used:  14289.5MB
- max. memory (heap) available: 14336MB

Note:
The Java heap size can be specified with the -Xmx option.
E.g., to use 128MB as heap size, the command line looks like this:
   java -Xmx128m -classpath ...
This does NOT work in the SimpleCLI, the above java command refers
to the one with which Weka is started. See the Weka FAQ on the web
for further info.
```

max. Size 100,000 currently: 1604 ☑ Use wordwrap

Figure 13: Out of memory error caused by training Weka on the full training dataset.

4.7 Implementation

4.7.1 Tools

Graphical visualization and database manipulation tools for data analysis

- XAMPP PhpMyAdmin web interface
- WEKA version 3.8.5 Workbench
- MySQL Client on CMD
- MySQL Workbench

Web scraper implementation kit

- VS Code IDE
- MySQL
- MySQL WorkBench, MySQL client and PhPMyAdmin for DB administration
- Sequelize Object Relational Mapper
- Node JavaScript runtime environment
- NPM package manager for Node
- Axios HTTP Client
- Cheerio DOM navigation and manipulator library

Dataset manipulation

- Python 3.8 environment
- Pandas
- PyCharm IDE
- Numpy

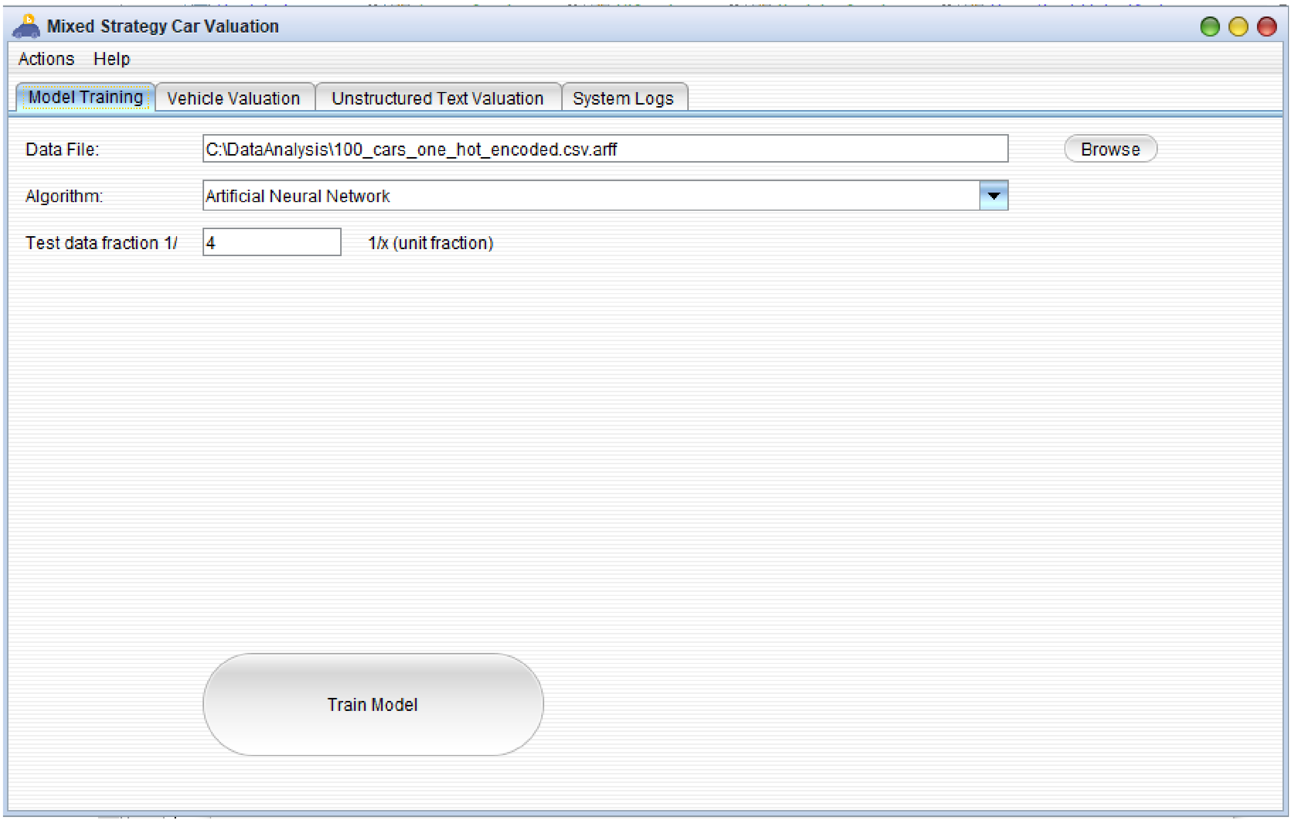
Prototype implementation

- JDK version 8+
- NetBeans IDE
- NetBeans Graphical User Interface Builder
- OpenNLP toolkit
- H2 in-memory persistence library
- Java Agent Development Framework
- Dl4jMlpClassifier for regression implementation based on deep learning

4.7.2 Graphical User Interface Design

The prototype's user interface was created using Java Swing, with the exact design created using the Netbeans Swing Builder tool.

For the prototype being constructed, an interface was created to accomplish each of the three key tasks: instance valuation from free text, instance valuation from structured data and model training. An additional tab is available to offer logs for diagnostics.

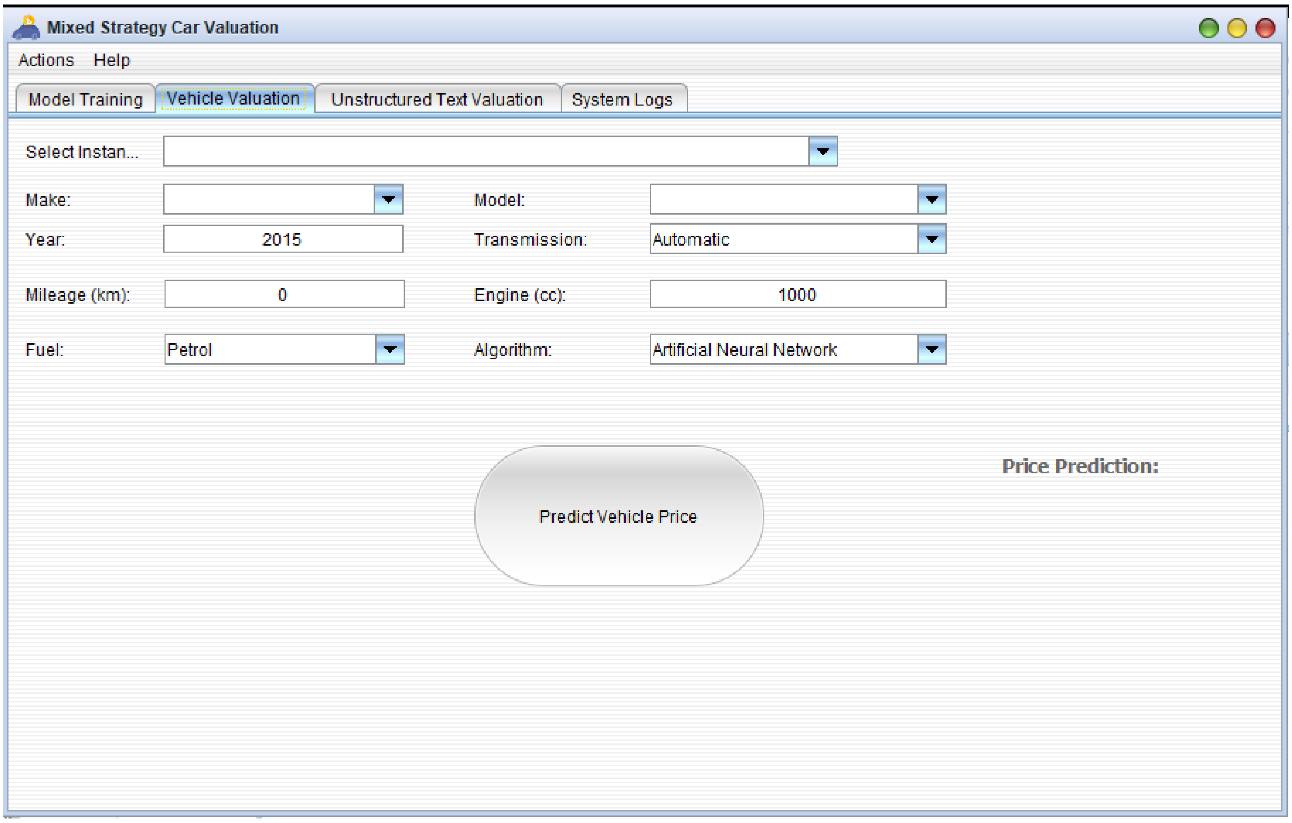


Data File: C:\DataAnalysis\100_cars_one_hot_encoded.csv.arff

Algorithm: Artificial Neural Network

Test data fraction 1/ 4 1/x (unit fraction)

Figure 14. Prototype UI: Model Training Tab



Select Instan...

Make:

Model:

Year: 2015

Transmission: Automatic

Mileage (km): 0

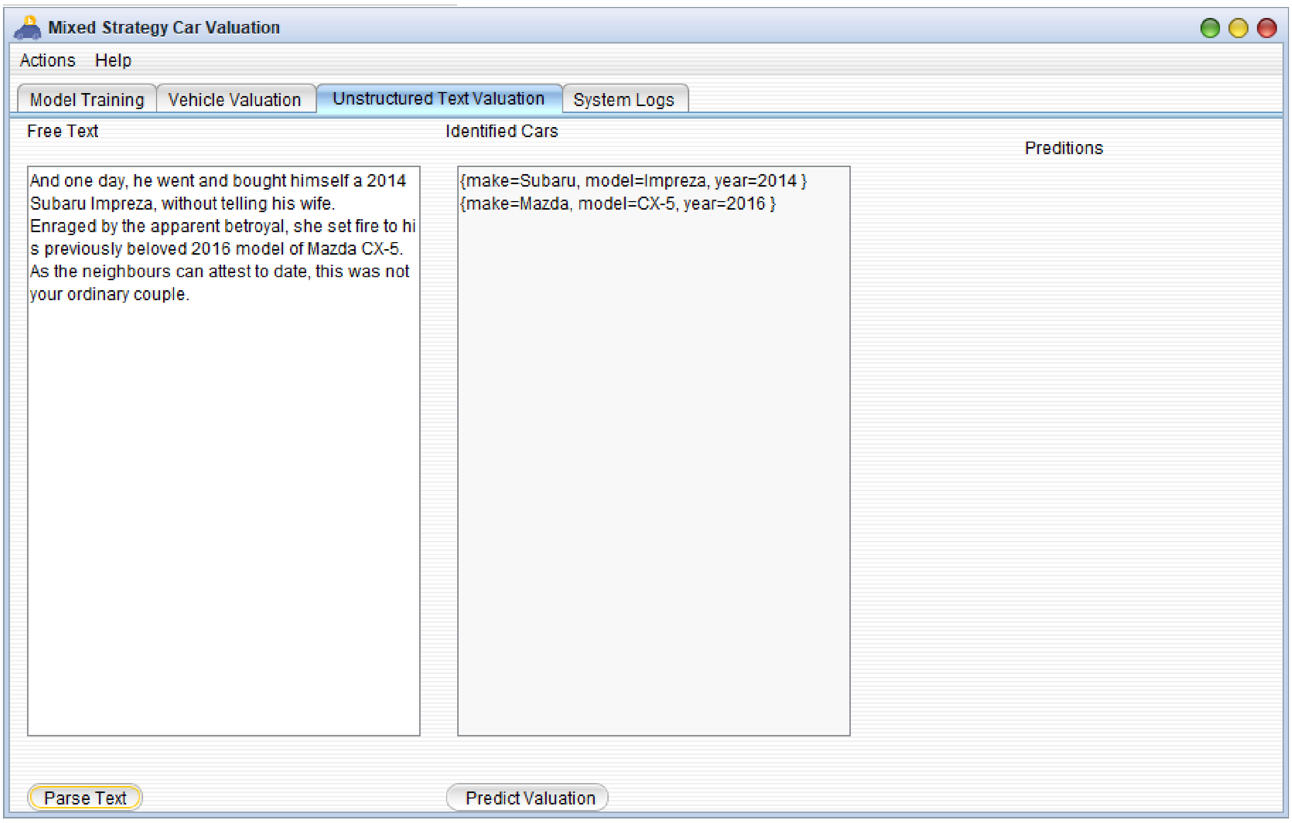
Engine (cc): 1000

Fuel: Petrol

Algorithm: Artificial Neural Network

Price Prediction:

Figure 15. Prototype UI: Prediction Tab



Free Text

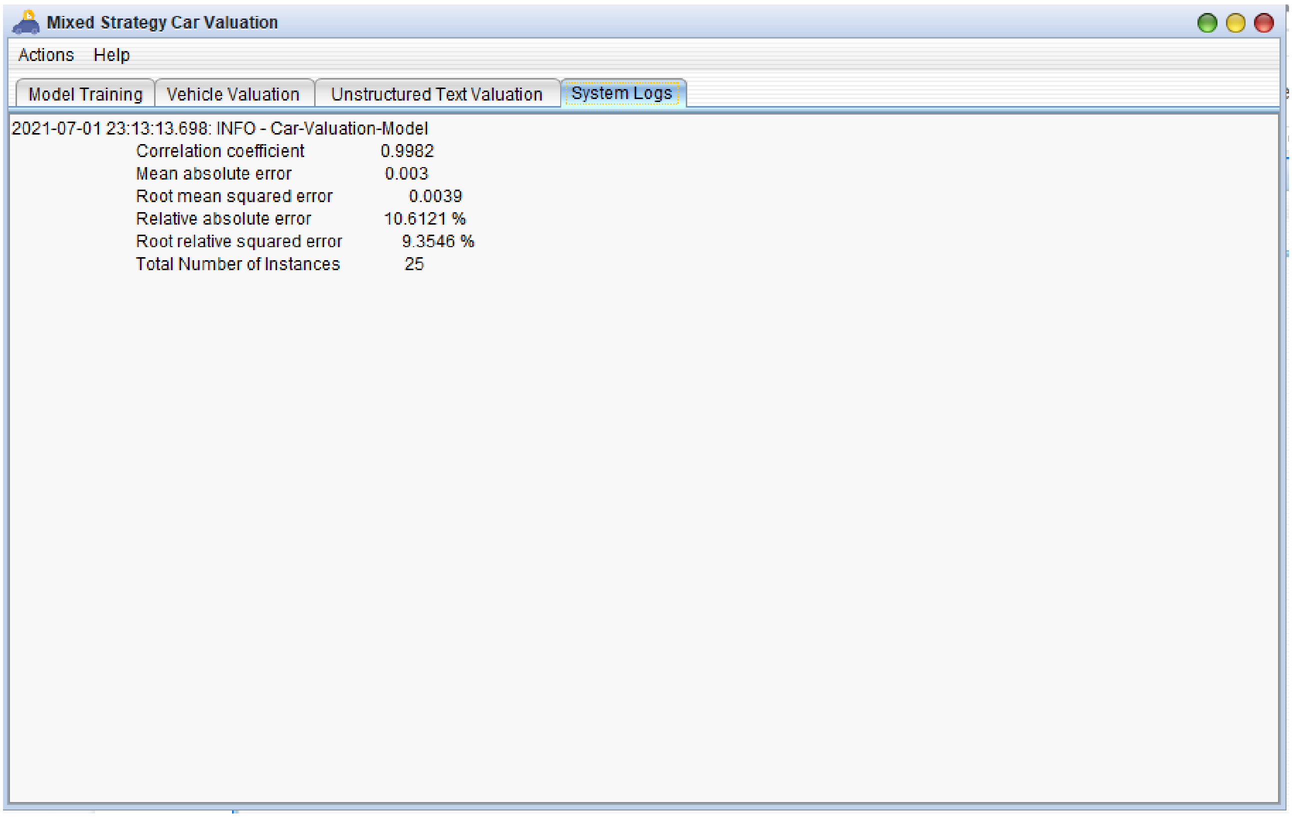
And one day, he went and bought himself a 2014
Subaru Impreza, without telling his wife.
Enraged by the apparent betroyal, she set fire to hi
s previously beloved 2016 model of Mazda CX-5.
As the neighbours can attest to date, this was not
your ordinary couple.

Identified Cars

```
{make=Subaru, model=Impreza, year=2014 }
{make=Mazda, model=CX-5, year=2016 }
```

Preditions

Figure 16. Prototype UI: Text Feature Extraction and Prediction Window

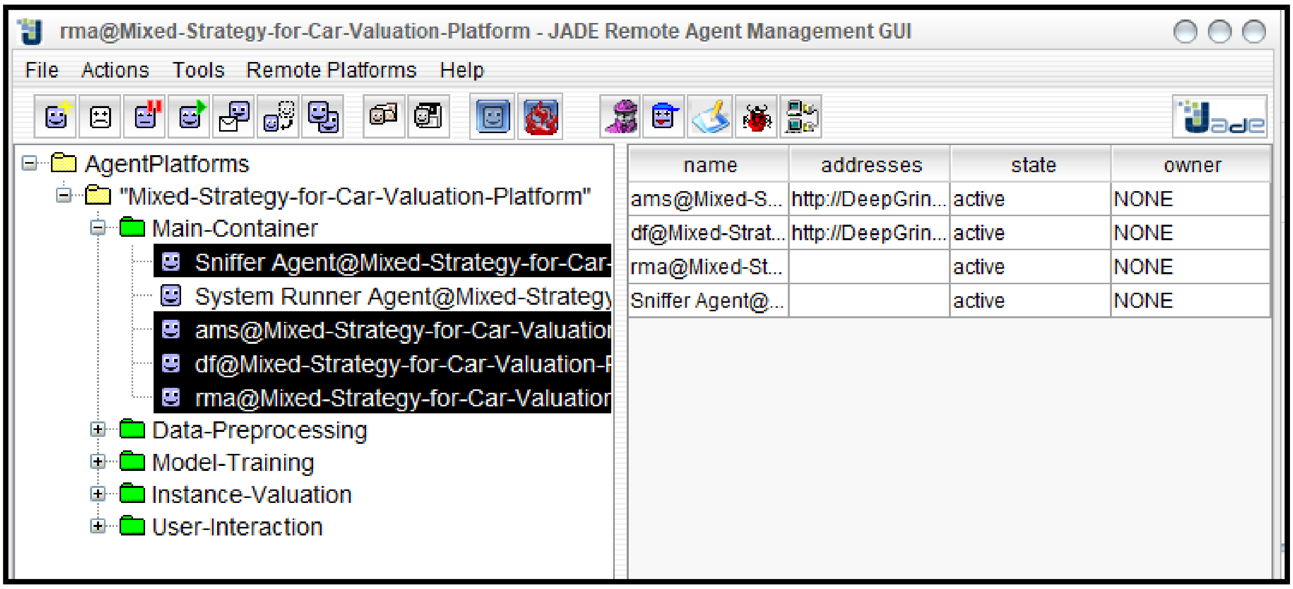


```
2021-07-01 23:13:13.698: INFO - Car-Valuation-Model
                  Correlation coefficient          0.9982
                  Mean absolute error               0.003
                  Root mean squared error             0.0039
                  Relative absolute error          10.6121 %
                  Root relative squared error        9.3546 %
                  Total Number of Instances          25
```

Figure 17. Prototype UI: System Logs Tab

4.7.3 Agent Platform

The Java Agent Design Framework was used to implement the agent platform used in this study. An agent container was created to house agents performing related tasks, with all other agents living in the Main-Container.


rma@Mixed-Strategy-for-Car-Valuation-Platform - JADE Remote Agent Management GUI
File
Actions
Tools
Remote Platforms
Help
AgentPlatforms
"Mixed-Strategy-for-Car-Valuation-Platform"
Main-Container
Sniffer Agent@Mixed-Strategy-for-Car-
System Runner Agent@Mixed-Strategy
ams@Mixed-Strategy-for-Car-Valuatior
df@Mixed-Strategy-for-Car-Valuation-F
rma@Mixed-Strategy-for-Car-Valuatior
Data-Preprocessing
Model-Training
Instance-Valuation
User-Interaction

name	addresses	state	owner
ams@Mixed-S...	http://DeepGrin...	active	NONE
df@Mixed-Strat...	http://DeepGrin...	active	NONE
rma@Mixed-St...		active	NONE
Sniffer Agent@...		active	NONE

Figure 18. Remote Monitoring Agent URI showing the main-container

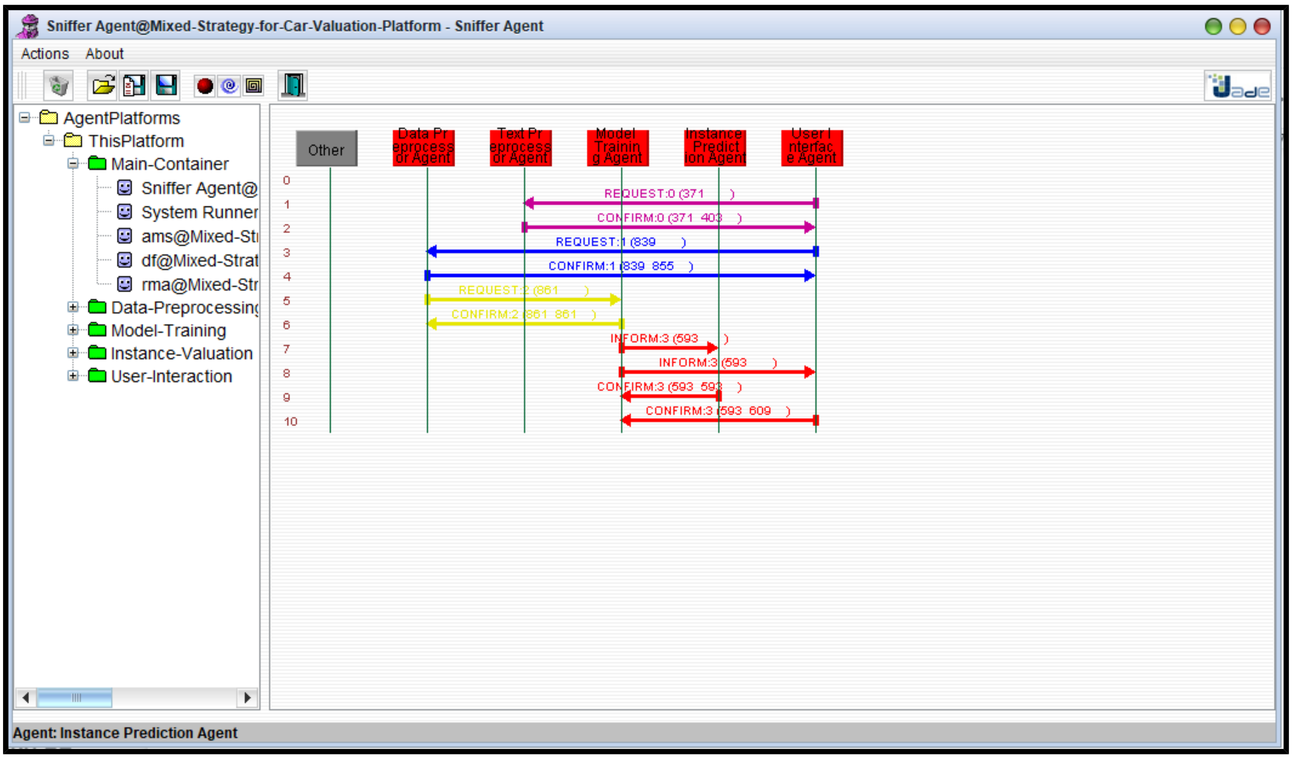


Figure 19. Communication between agents shown by a Sniffer Agent

Chapter 5: Results and Discussion

This chapter presents the results of the project's model comparison and explores how they relate to the project's goals.

5.1 Results

The Artificial Neural Network (multi-layer perceptron), the SMOreg implementation in WEKA 3.8.5, and the Deep Learning-based regression given by WekaDeeplearning4j were all used to create and direct three models. Each algorithm was taught on the homogeneous data and then analysed and examined on similar datasets to compare their performance in terms of corresponding and error rates. For each of the datasets with different selected properties, the performance of the various algorithms on the test dataset is shown in the tables below.

Because this is a regression problem, measures like accuracy, precision, recall, and F-measure can't be utilized to assess the models' performance. Instead, we look at how close the predicted values are to the true values by running tests and evaluation sets against the trained model and using the deviation to determine input variable correlation to the predicted value, as well as different sorts of errors that influence model performance.

	Bivariate correlation	Average absolute error	RRSE	Relative absolute error (RAE)
ANN MLP	0.7968	0.0042	29.1771%	30.3192%
SVM reg	0.8362	0.0017	25.1862%	20.0801%
Dl4jMlpClassifier	0.7301	0.0048	28.3412%	33.0153%

Table 16. Algorithm performance: 4433 rows, 3 features.

	Bivariate correlation	Average absolute error	RRSE	Relative absolute error (RAE)
ANN MLP	0.8027	0.0142	69.8337%	60.6158%
SVM reg	0.6297	0.0398	77.3236%	83.1387%
Dl4jMlpClassifier	0.9977	0.0036	13.6785%	11.5543%

Table 17. Algorithm performance; 500 rows, 1000 training cycles

	Bivariate correlation	Average absolute error	RRSE	Relative absolute error (RAE)
ANN MLP	-	-	-	-
SVM reg	-	-	-	-
Dl4jMlpClassifier	0.1475	0.0145	31.4291 %	40.8804 %

Table 18. Algorithm performance; all attributes, 10 training cycles

5.2 Discussion

Through this research, the researcher was able to meet all the set objectives for the research and answer the questions raised during the proposal stage for. The premier objective was to find the current practices used to evaluate vehicles. The contemporary practice was found to be based principally on reliance on expert opinions and mathematical formulae that calculate the compound annual depreciation and subtract it from the buying price of the car, adjusting for annual inflation, to get the residual value of vehicles. The other objectives were to design, implement and evaluate a mixed strategy prototype.

Upon reviewing work done by other researchers on this problem, and also in other domains of related data profiles, the researcher established that regression algorithms provided a reliable option for vehicle valuation, and thus was able to design a prototype based on 3 regression algorithms, namely SMOreg, MultiLayer Perceptron and Deep Neural Networks. SMOreg gave the best results for a small dataset while the Dl4jMlpClassifier gave the least error margins overall with 11.5543% relative mean error in the final model embedded in the functional prototype.

Given the temporal, budgetary and computational resource restrictions on this study, there is great potential for improving the performance of the final model given more time, data and computing power.

In the following sub-sections, the researcher discusses some of the challenges faced and insights acquired while conducting the research that future researchers have to consider while embarking on similar work.

5.2.1 Quality of data

One of the biggest challenges faced was the non-uniformity of data. Web scraping from 22 different car dealers sites was a data source, the data from each was missing some features be it by design or mistake. It is not ideal to train models on data with about only 10% of the dataset containing all the required features.

Collecting feature data about known vehicle models by year, and using it to populate some of the missing attributes could increase the usable data increased to 25% without compromising the accuracy of the data.

5.2.2 Nature of data

Vehicle data is majorly categorical. Of the initial 14 attributes in consideration only 7 are numeric values; year, price, mileage, engine, number of doors, airbags and seats and of these, only mileage has a lot of variation.

The categorical attributes cannot be fed into a regression algorithm in nominal form, but they have to be converted into a sparse matrix of binary values which expands the dataset horizontally, increasing the number of attributes exponentially. Each different categorical value translates to a new attribute for all the instances in the data.

On performing one-hot encoding of the 54,000 instances selected to train the models, the resulting dataset had more than 4000 attributes, and it grew in size from 6 MB to occupy 3.2 GB on the disk,

written in CSV format.
This increased the computational resources required to train the data to exceed those at the disposal of this research.

5.2.3 Dimensionality reduction

Having data with more than 4000 attributes is not ideal for training a machine learning model especially with constraints of time and computing resources. Using Principal Component Analysis, the dataset was reduced to 47 attributes while keeping 95% of the variance contained in the original dataset.
The original attributes were factored with eigenvalues determined by PCA analysis to create the new attributes.
While this gave a more manageable dataset, the regression models based on SMOreg and ANN could not converge on this dataset after running for more than a week each.
And again, when done programmatically, the eigenvalues are not available to transform new instances for prediction.

5.2.4 CPU vs GPU

The final algorithm selected, Dl4jMlpClassifier, was a deep learning algorithm, which could utilize the GPU (GeForce GTX 1060 Graphics Cards) to reduce compute times considerably and train on the full dataset within 4 days, albeit for only 10 iterations.
This is a big improvement as training the same algorithm on the CPU, for the same parameters, on the same training data was projected to take 110 days.

5.2.5 Suitability of the Multi-agent Architecture

Throughout this research, the power of multi-agent systems has been evident. The modular approach it enforces on the software design and implementation, and the potential for interaction, learning and mobility presented by agents has the potential to simplify the creation of complex systems made of thousands of independent and interchangeable parts.
However, the potential of multi-agent architecture is underutilized in this research. The nature of the problem, the data and the algorithms used means a monolithic architecture would have suited the proposed design just as well, with the added benefit of reducing the time taken to fabricate the final prototype.

5.2.6 Ethical Considerations for Collecting Data by Web Scraping

Using bots to crawl a website to collect data can have serious effects on the online resource, resulting in a denial-of-service attack and making the resource unavailable to the intended target, and costing the owners potential customers, income and money to perform maintenance on their servers.
As such, before embarking on scraping, it is advisable to contact the host of the required data and request it.
If web scraping is the only way to get this data, then the design of the bots should take care to avoid

continuous and parallel requests, taking proportional pauses between requests to allow the server to handle requests from other users.

It would also be considerate to avoid peak visit hours of the site, even running the bots during night hours when the majority of the service’s target population is asleep.

Finally, it is advisable to include contact information in the request headers to let the system administrators contact you in case of emergencies that could necessitate halting the scraping process.

The web scraping bots utilized in this research all observed these guidelines.

Chapter 6: Conclusion and Recommendations

6.1 Conclusion

Machine learning algorithms can provide a reliable option to the traditional formula and expert opinion-based approach currently used for vehicle valuation in Kenya.

For the perfect AI solution, it is necessary to perform a collection of quality data over an extended period and to create models without limits in time, finances or computing resources.

And while it might be costly to train the models, once ready, the prediction will happen within milliseconds. One model could perform more predictions in a second than an expert armed with a calculator or a spreadsheet with formulas can make in a year.

This would also mean that vehicle valuation is accessible to more people, and those who deal with experts have a quick way to get a second opinion.

6.2 Contributions

The methodology, experiments and findings of this research contribute to the body of knowledge in the domain of vehicle valuation in Kenya. Having successfully applied different supervised machine learning algorithms namely Multilayer Perceptron, SMOreg and GPU based Deep Learning to determine the price of vehicles, this research has paved the way for future research using these algorithms or others to improve on the solution.

Researchers who embark on similar or related work will be aware of the challenges that await with the data collection, data pre-processing and training the models and thus can find ways to work around these challenges to improve on the solution.

The final prototype is fully functional and can be used to perform the prediction of vehicle values from both structured and unstructured data. Structured data is first extracted from textual data and processed from thereon in the same manner as structured data.

6.3 Future Work

This research provides a good basis for future work in similar and related domains.

One of the challenges to overcome is the availability of quality data as in the end, only 25% of the data collected in this research contained all of the required attributes to be used, even after concerted efforts to plug in the missing gaps.

Having a lot of training data without all the required attributes also means that the end-user of the prediction models may not have all the attributes at the time of determining the value of their vehicle. This means that the end system should either be able to work without some given attributes, either by not including them in the model or by having a way to estimate them. This could be by selecting the mean or mode from available training data or designing a whole new system to handle this task.

A sizable vehicle dataset is likely to contain a lot of different categorical values each of which would increase the number of attributes in the nominal to a binary converted dataset. This means some effort needs to go into dimensionality reduction so that the dataset can be scaled down vertically without losing a lot of the important information to scale down the computational requirements to train models for any algorithm with some constraint of time and budget. Lastly, there are more regression algorithms that this research did not test, and more will be developed in future. This opens the opportunity to research the same domain using different algorithms or improve on those used in this research.

References

1. Bennett, Victoria J. "Effects of road density and pattern on the conservation of species and biodiversity." Current Landscape Ecology Reports 2.1 (2017): 1-11.
2. Chowdhery, Syed Azad, and Marco Bertoni. "Modeling resale value of road compaction equipment: a data mining approach." IFAC-PapersOnLine 51.11 (2018): 1101-1106.
3. Holmes, Geoffrey, Andrew Donkin, and Ian H. Witten. "Weka: A machine learning workbench." Proceedings of ANZIIS'94-Australian New Zealand Intelligent Information Systems Conference. IEEE, 1994.
4. Chang, Jiangxue, and Yuquan Zhang. "Research on Second-hand Vehicle Evaluation System Based on Improved Replacement Cost Method." (2018): 1290-1293.
5. Merton, Robert C. "Theory of rational option pricing." The Bell Journal of economics and management science (1973): 141-183.
6. Wooldridge, Michael, and Nicholas R. Jennings. "Intelligent agents: Theory and practice." The knowledge engineering review 10.2 (1995): 115-152.
7. Madan, Dilip B. "Asset pricing theory for two price economies." Annals of Finance 11.1 (2015): 1-35.
8. Gongqi, Shen, Wang Yansong, and Zhu Qiang. "New model for residual value prediction of the used car based on BP neural network and nonlinear curve fit." 2011 Third International Conference on Measuring Technology and Mechatronics Automation. Vol. 2. IEEE, 2011.
9. ÖZÇALICI, Mehmet. "Predicting second-hand car sales price using decision trees and genetic algorithms." Alphanumeric Journal 5.1 (2017): 103-114.
10. Kieti, Mutisya Raphael. Application of multiple regression analysis (MRA) in the Valuation of used motor vehicles. A case study of used saloon cars. Diss. 2005.
11. Riikkinen, Mikko, et al. "Using artificial intelligence to create value in insurance." International Journal of Bank Marketing (2018).

12. Arel, Itamar, Derek C. Rose, and Thomas P. Karnowski. "Deep machine learning-a new frontier in artificial intelligence research [research frontier]." IEEE computational intelligence magazine 5.4 (2010): 13-18.

13. Bonaccorso, Giuseppe. Machine learning algorithms. Packt Publishing Ltd, 2017.

14. Chaphalkar, N. B., and Sayali Sandbhor. "Use of artificial intelligence in real property valuation." International Journal of Engineering and Technology 5.3 (2013): 2334-2337.

15. Damodaran, Aswath. Investment valuation: Tools and techniques for determining the value of any asset. Vol. 666. John Wiley & Sons, 2012.

16. Du, Jie, Lili Xie, and Stephan Schroeder. "PIN optimal distribution of auction vehicles system: Applying price forecasting, elasticity estimation, and genetic algorithms to used-vehicle distribution." Marketing Science 28.4 (2009): 637.

17. Ifthikar, Amjadh, and Kaneeka Vidanage. "Valuation of Used Vehicles: A Computational Intelligence Approach." 2018 8th International Conference on Intelligent Systems, Modelling and Simulation (ISMS). IEEE, 2018.

18. Michalek, Jeremy J., et al. "Valuation of plug-in vehicle life-cycle air emissions and oil displacement benefits." Proceedings of the National Academy of Sciences 108.40 (2011): 16554-16558.

19. Mora-Esperanza, Julio Gallego. "Artificial intelligence applied to real estate valuation: An example for the appraisal of Madrid." Catastro, April 1 (2004): 255-265.

20. Morano, P. I. E. R. L. U. I. G. I., F. R. A. N. C. E. S. C. O. Tajani, and CARMELO MARIA Torre. "Artificial intelligence in property valuations: an application of artificial neural networks to housing appraisal." Advances in Environmental Science and Energy Planning (2015): 23-29.

21. Pal, Nabarun, et al. "How much is my car worth? A methodology for predicting used cars' prices using random forest." Future of Information and Communication Conference. Springer, Cham, 2018.

22. Pudaruth, Sameerchand. "Predicting the price of used cars using machine learning techniques." Int. J. Inf. Comput. Technol 4.7 (2014): 753-764.

23. Rowell, Arden. "Partial valuation in cost-benefit analysis." Admin. L. Rev. 64 (2012): 723.

24. Shayamali, P. P. D. N. A Study on the Current System of Vehicle Valuation in Sri Lanka. Diss. University of Sri Jayewardenepura, Nugegoda, 2013.

25. Smith, Tony C., and Eibe Frank. "Introducing machine learning concepts with WEKA." Statistical genomics. Humana Press, New York, NY, 2016. 353-378.

26. Simeunović, N., et al. "Improving workforce scheduling using artificial neural networks model." Advances in Production Engineering & Management 12.4 (2017): 337-352.

27. Wu, Jian-Da, Chuang-Chin Hsu, and Hui-Chu Chen. "An expert system of price forecasting for used cars using adaptive neuro-fuzzy inference." Expert Systems with Applications 36.4 (2009): 7809-7817.

Appendices

Flow chart of machine learning centred research

A representation of the research process focused on the steps relevant to the machine learning process.

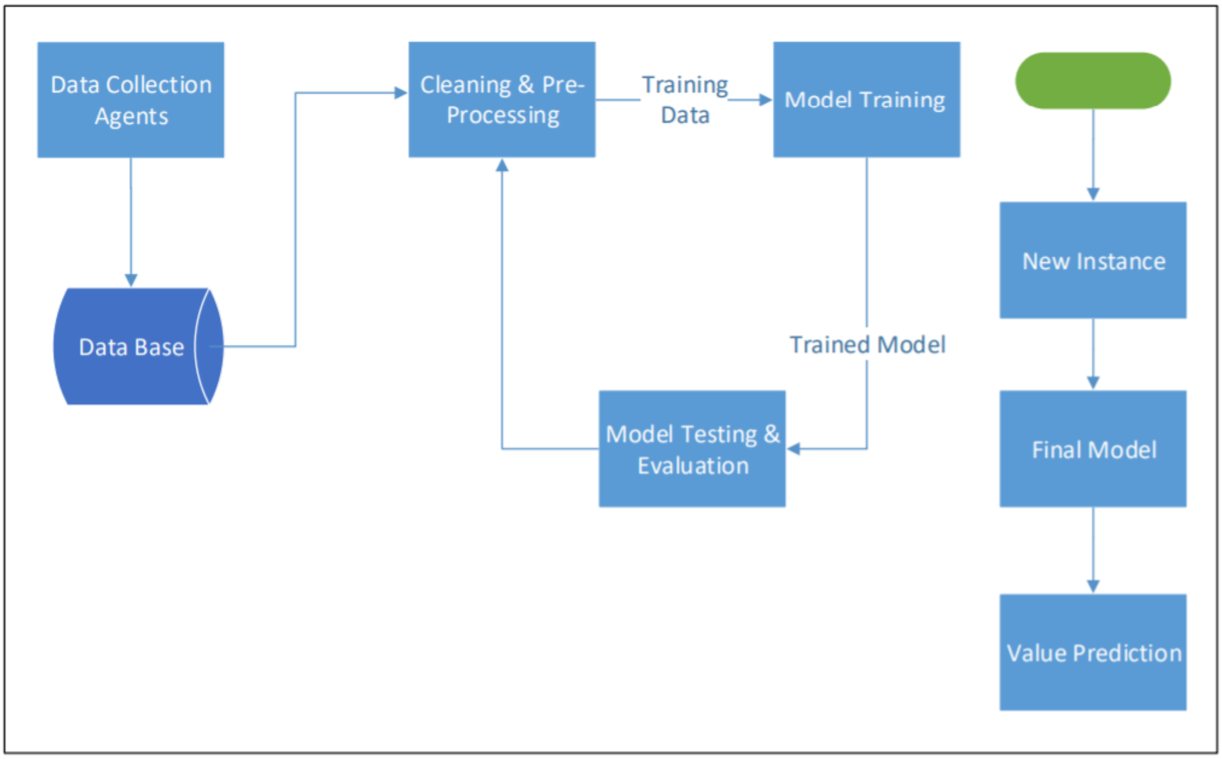


Figure 20. Research process, highlighting machine learning steps

Sample SQL queries deployed to cleanse unwanted data from the repository

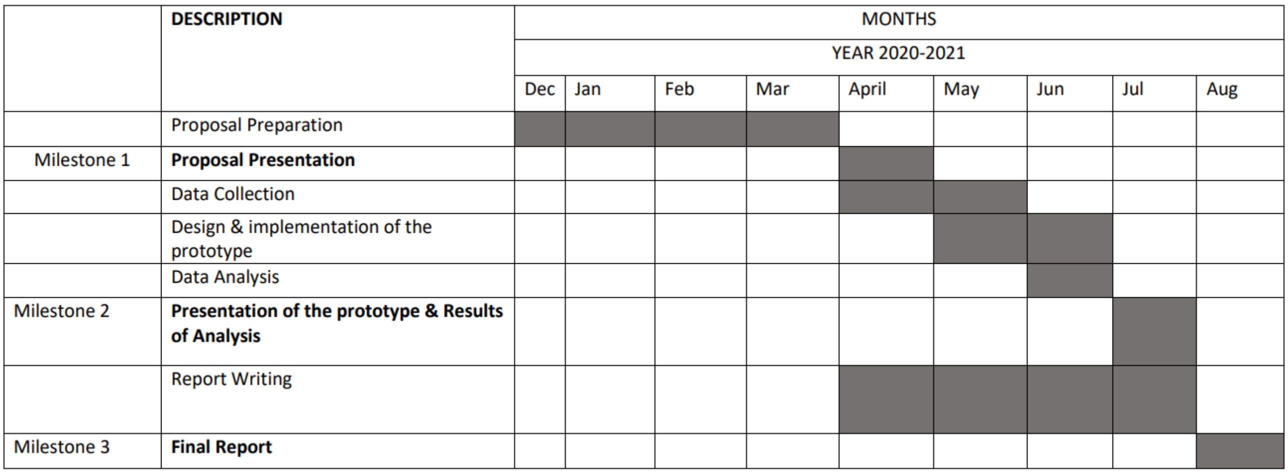
```
INSERT INTO `_cars_copy`(`title`, `id`, `origin`, `make`, `color`, `model`, `transmission`, `year`,
`fuel`, `price`, `engine`, `mileage`) SELECT `title`, `id`, `origin`, `make`, `color`, `model`,
`transmission`, `year`, `fuel`, `price`, `engine`, `mileage` FROM `cars` WHERE make is NOT null
and model is NOT null and model != '' and year is not null AND year>1900 and year<2022
```

```
SELECT AVG(price), `make`, `model`, `transmission`, `fuel`, `year`, `engine`, `mileage` FROM
cars_copy WHERE make IS NOT NULL AND LOWER(make) NOT REGEXP '^([[:space:]]*|
others*)$' AND model IS NOT NULL AND LOWER(model) NOT REGEXP '^([[:space:]]*|
others*)$' AND transmission IS NOT NULL AND LOWER(transmission) NOT REGEXP
'^([[:space:]]*|others*)$' AND fuel IS NOT NULL AND LOWER(fuel) NOT REGEXP
'^([[:space:]]*|others*)$' AND year IS NOT NULL AND year>=1970 AND year<=2021 AND
```

```
engine IS NOT NULL AND engine>=300 AND engine<=20000 AND mileage IS NOT NULL
AND mileage>=0 AND mileage<=250000 GROUP BY `make`, `model`, `transmission`, `fuel`,
`year`, `engine`, `mileage` HAVING count(*)>=3

SELECT count(*) as totals, AVG(price) as target, year, make, model FROM `cars_copy` WHERE
make is NOT NULL AND year IS NOT NULL and model is NOT null and year>1960 and
year<2022 and model!= " and make != " GROUP by make, model, year HAVING totals>=5
ORDER by totals, make, model, year

SELECT count(*) as totals, AVG(price) as target, year, make, model FROM `cars_copy` WHERE
make is NOT NULL AND year IS NOT NULL and model is NOT null and year>1960 and
year<2022 and model!= " and make != " AND (make, model) in (SELECT make, model FROM
`cars_copy` WHERE make!=" and make is NOT null GROUP BY make, model HAVING count(*)
>= 10) GROUP by make, model, year HAVING totals>=5 ORDER by totals, make, model, year
```

Approximate project schedule

Successful completion of the study is expected to adhere to the schedule shown in the below Gantt chart, only with slight deviations.



	DESCRIPTION	MONTHS								
		YEAR 2020-2021								
		Dec	Jan	Feb	Mar	April	May	Jun	Jul	Aug
	Proposal Preparation	▨	▨	▨	▨					
Milestone 1	**Proposal Presentation**					▨				
	Data Collection					▨	▨			
	Design & implementation of the prototype						▨	▨		
	Data Analysis							▨		
Milestone 2	**Presentation of the prototype & Results of Analysis**								▨	
	Report Writing					▨	▨	▨	▨	
Milestone 3	**Final Report**									▨

Figure 21. Expected project duration

Requirements

Physical resources:

1. Powerful portable computer
2. Graphical Processing Unit to expedite algorithm training

Open-source software resources:

1. JDK version 8+
2. Netbeans IDE
3. Waikato Env. for Knowledge Analysis 3.8.5
4. Waikato Env. for Knowledge Analysis Deeplearning4J

5. JADE Framework for MAS implementation
6. Python Environment version 3.8
7. Pandas library
8. Numpy library
9. Google TensorFlow
10. MySQL relational database
11. Nvidia Cuda toolkit

Financial plan

Product	Amount	Expenditure KSh
Logistics and records	~	20000
Computer	1	100000
Supplemental GPU	1	70000
Open-source software	~	0
Total outlay		**190000**

Table 19. Expected costs