Car Price Prediction using Machine Learning Techniques

Catalin Alexandru Mihalache
986965
Supervisor: Matt Roach

May 2022

Submitted to Swansea University in fulfilment
of the requirements for the Degree of Bachelor of Science

Bachelor of Science

Swansea University
Prifysgol Abertawe

Department of Computer Science
Swansea University

May 3, 2022

Declaration

This work has not been previously accepted in substance for any degree and is not being concurrently submitted in candidature for any degree.

Signed  (candidate)

Date **03/05/2022**

Statement 1

This thesis is the result of my own investigations, except where otherwise stated. Other sources are acknowledged by footnotes giving explicit references. A bibliography is appended.

Signed  (candidate)

Date **03/05/2022**

Statement 2

I hereby give my consent for my thesis, if accepted, to be made available for photocopying and inter-library loan, and for the title and summary to be made available to outside organisations.

Signed  (candidate)

Date **03/05/2022**

Abstract

A car price prediction has been a high-interest research area, as it requires noticeable effort and knowledge of the field expert. Therefore, many distinct attributes are examined for reliable and accurate predictions.
In this dissertation, we investigated the application of supervised machine learning techniques to predict the price of used cars in the United Kingdom. The predictions were based on data collected from the PistonHeads website using our custom web scraper. Different techniques like linear regression, extreme gradient boosting, random forest, and neural networks have been used to make predictions. The results were then evaluated and compared to find the model with the best performance. Gradient boosting proved to have the best performance by scoring the lowest errors of the four models. The conclusion was drawn that although XGBoost was better for our small custom dataset, the deep learning approach has the most potential for a higher accuracy on a larger dataset.

Table of Contents

1 Introduction

Car price prediction is both essential and exciting at the same time. The number of registered cars in the UK was 32,884,320 [1], nearly 8% more than in 2014. Data from the last two years is irrelevant as the coronavirus pandemic influenced the car market more than any other factor. With the economic problems related to the pandemic, it is more than likely that second-hand and used cars sales will increase. On the other hand, the new car market declined by 2.4% in 2019 [2], with the annual registrations of new cars diminishing for the 3rd year in a row.

In the UK, car leasing has become very popular. Over 1.6 million people are driving lease cars, and the growth rate for 2019 was 14% [3]. Leasing is an option for financing a good. The owner of the good is called the lessor, while the person who rents the good is called the lessee. The lessee must pay fixed instalments for several months/years agreed by both parties in exchange for renting the goods. At the end of the leasing period, the lessee has an option to buy the good at its residual value. Hence, the lessor must predict the salvage value with accuracy. If the lessor overlooks the residual value, the monthly payments will be more outstanding than expected, and clients will look for another financer.

On the other hand, if the residual value is bloated, the clients will benefit by paying lower instalments. However, the lessor will not sell the goods at the overpriced value, so they will be on loss rather than a profit. For example, in 2008, car manufacturers from Germany registered a 1-billion-euro loss because of residual value misinterpretation [4]. This fact adds significance to the problem of car price prediction.

Car price prediction is not a straightforward project. It involves expert knowledge, as the price depends on multiple features and factors. Usually, the most important ones are car make, model, car age, mileage, and car colour. The number of features is not fixed, as there can be a lot of valuable features. According to a 2020 Statista survey [5], the most critical feature for British consumers was fuel efficiency. Vehicle safety, suitability for everyday use, low price, and high driving comfort were the features rounding off the top five.
This study explores some machine learning techniques to foresee the price of used cars with maximum accuracy. The algorithms used were linear regression, decision trees, random forests, and deep learning. In addition, evaluation metrics were applied to each model, and lastly, a comparison was made to determine the best approach.

1.1 Motivation

One of the general aims in the modern world of technology is to predict the future as accurately as possible. Moreover, substantial social and economic changes can affect the world, and most countries rely on technological development [6]. Therefore, by having an accurate forecast of the future, governments will have more time to make plans to prevent problems. The primary approach to this is using machine learning methodologies, which have been applied in this study. Concretely, an artificial intelligence (AI) program simulates how the human brain works, trains, and learns from mistakes to improve predictions.

The prediction problem is challenging, yet it is necessary to impact car leasing businesses' marginal profit. Therefore, the purpose of this study is to predict car prices accurately. The supervised learning technique was applied first due to the high probability of the input being labelled. Supervised learning's essential task is to map the input features to the output label. Supervised learning is a subcategory of machine learning. Due to the satisfactory results, applying reinforcement learning was not needed.

1.2 Project Aims

This study is briefly presenting the following aspects:

- To explore the machine learning subject and understand how it works
- To identify the web scraping challenges and risks
- To build a car price prediction model which has a high accuracy
- To produce specification and evaluation criteria

This project is about building a car price prediction model. If it is accurate, this could find its usage in the car leasing businesses. It would help them predict the salvage value of the car they leased. Thus, it ensures profit for the company and a fair price of monthly instalments for the client. Supervised learning was the preferred method, with reinforcement learning being the second option. Comparisons between algorithms were made to outline possible differences or similarities, which helped better understand how machine learning works. Thus, code was written in Python 3.0 using specific machine learning libraries like TensorFlow and Keras.

Moreover, the data for the study was extracted using a web scraper. The web scraper allows a collection of a large amount of data. The downside of this method is that many samples will not have all the fields required. So, only a small subset of features will be considered. In addition, the scraper was written so that websites did not IP block the computer from requesting too many resources. The web scraper used BeautifulSoup and Request libraries.

The evaluation metrics used were mean square error (MSE), mean absolute error (MAE), mean absolute percentage error (MAPE), and mean squared logarithmic

error (MSLE). Finally, a thorough evaluation of the results was made to see the reliability of each model.

1.3 Contributions

- **Implementation of a Web Scraper**
  We successfully designed and implemented a web scraper to get our custom dataset used for training and testing the model. The program scrapes all the features needed, only filling the dataset with empty values when the data is missing.

- **Implementation of Supervised learning models**
  After the data collection was finished, we implemented various models. We applied linear regression, extreme gradient boost regressor (XGBoost), random forest, and deep learning.

- **Comparative analysis of various ML models**
  We successfully compared the solutions and studied their performances and characteristics in our environment. For instance, one critical success we achieved was that sometimes clustering could outperform deep learning models if the input data is labelled.

1.4 Project Outline

This paper is organized in the following manner: Section 2 contains related work in machine learning and its subsidiaries and previous studies on price prediction. In Section 3, the study's methodology, alongside the programming languages, tools, and frameworks we used, are illustrated. Section 4 describes the experiments we carried out in detail, their results, and our evaluation and analysis. The management of this project will be discussed in Section 5. Finally, in Section 6, reflections on the outcome alongside potential future works and a summary are given.

2 Background research

Although machine learning has been introduced for over half of a century now, more precisely in 1959 by Arthur Samuel [7], the benefits it brings are not yet known to the ordinary man. However, companies have been willing to invest in machine learning projects more than ever in the last years. According to Forbes's article, in 2019, over $50 billion was invested in AI systems [8].

2.1 Machine Learning

"Machine learning is programming computers to optimize a performance criterion using example data or past experience" [9]. Before the execution process, a model is defined with already known parameters. A model can be predictive to predict the future or descriptive to improve its knowledge from the training data. The learning process is the execution of a program. After each execution, the model parameters are updated to more reliable values than before using the training data. In the training phase, efficient algorithms are needed to get the optimal values for the model parameters. Generally, a heavy amount of data needs to be stored and processed. Hence, some people refer to machine learning as "Big Data" for its capacity to store the data needed. A diagram for the general process of machine learning can be seen below.

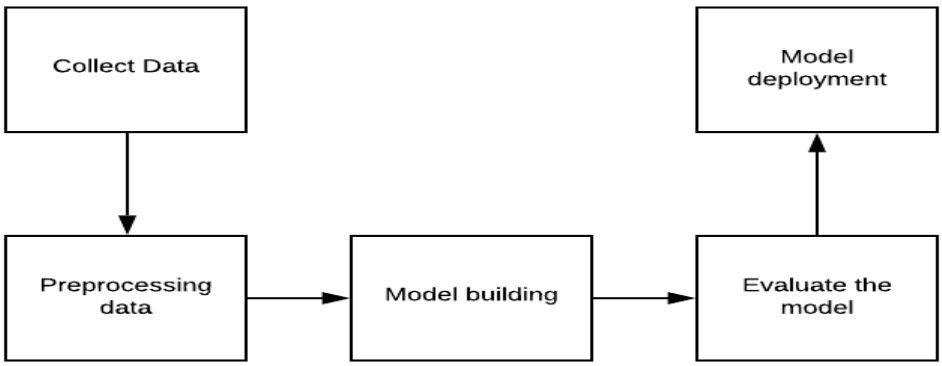

Diagram 1 – General process of machine learning

A machine learning model can be of two types: parametric and non-parametric. If a model is parametric, it has a fixed number of parameters. This model is faster to use but makes strong assumptions about the nature of the data distribution [10]. If the number of parameters grows with training data, the model is non-parametric. It is more flexible than parametric models, but it cannot be used on large datasets due to its slower speed.

Other concepts of machine learning are overfitting and underfitting. Those are measurements of model performance. Overfitting refers to a model that models the training data too well [11]. It happens when a model learns the detail and noise in the training data to the extent that it impacts its performance on unseen data. To limit the

overfitting, one can use data shuffling on the training data and hold back a dataset for testing. On the other hand, underfitting means that the model neither models the training data nor generalizes new data [11]. If this happens, another algorithm should be used. Below, a graphical interpretation of these concepts is shown.

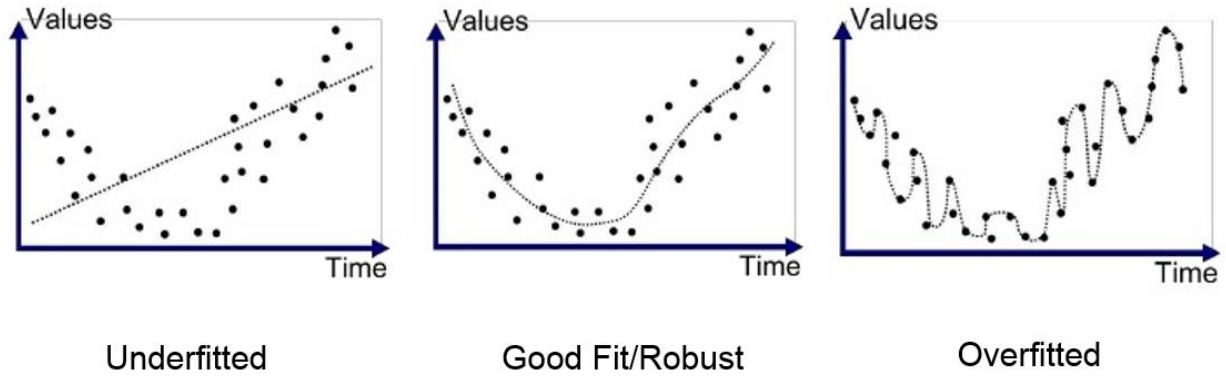

Figure 1 - Graphical interpretation of overfitting and underfitting [12]

There is no general model that can work effectively for all kinds of problems [10]. This scenario is the so-called no-free lunch theorem, and Walpert introduced it in 1996. The main reason behind this is that each domain needs to have different assumptions. An assumption that works in one domain may not work in another.

2.2 Supervised Learning

If the data is labelled, then the learning is called supervised. "Supervised learning is a machine learning methodology for creating a function from training data" [13]. Predicting the value after seeing several examples is the task of supervised learning. The overall aim is to generalize training data to unseen conditions as accurately as possible. The training set has both input and output objects. The output can predict a class label of the input.
There are a few types of supervised learning [14]:

- Regression – from the training data, a probabilistic output value is generated. After weighing the input variables, this gets a final value. This method can flop when handling nonlinear and multiple decision boundaries.
- Classification – as its name shows, it groups the data into classes. Multiple classification occurs when the data is split between more than two classes.
- Neural networks – is the most demanding technique. It requires powerful computational resources and is usually used in pattern recognition. It might not be a good choice if the input is too large.
- Support vector machines – is the most advanced technology in the category of supervised learning. It is classified as a discriminative classifier.

2.3 Artificial Neural Networks

"ANNs are computational models that can mimic the way nerve cells work in the human brain" [15]. They are often used in the case of nonlinear datasets for their capacity to adjust weights after each step. This is called the Backpropagation method. The steps for Backpropagation are [16]:

- Train the data and compute an output
- Calculate the error between the output and the actual value
- Adjust the weights
- If the error is higher than the predefined tolerance, it goes back to step 1; if not, stop.

This approach can be very efficient for the research. However, finding the correct weights for each feature is not very easy. For example, multiple customers may have different measurements when choosing which car to buy. Trying different weights can help find hidden insights about each feature.
Advantages of ANNs are [17]:

- Ability to work with incomplete knowledge
- Having a fault tolerance
- Is storing information on the entire network

Disadvantages of ANNs are [17]:

- Hardware dependent
- The duration time of the process is unknown

2.4 Related Work

Predicting the price of used cars has been studied intensively over the last decade. As a result, it is well known that it has become an exciting area of research for many scientists and machine learning enthusiasts.

In a Master thesis written by Listiani [4], she showed the strength of Support Vector Machines (SVMs) in price prediction. She built a regression model using SVM, and the precision was higher than other approaches like multivariate regression and simple multiple regression. This strengthens the idea that SVM is better on large datasets and has a lower probability of overfitting or underfitting. However, a downside of this study is that SVM superiority has not been shown in indicators such as mean, variance, or standard deviation.

Richardson [18], also in his Master's thesis, had a theory that in the last years, car manufacturers have been developing more durable cars. His study proved that hybrid cars could retain their value more than traditional cars using multiple regression analysis. The study has its origin from the environmental concerns that multiplied in the last years. With those concerns in sight, producers give their cars higher fuel efficiency.

Moreover, in his study, Pudaruth [19] applied machine learning algorithms to predict the car price in Mauritius. He manually collected data from a local newspaper, which can be a drawback as there is only a limited amount that he could get. As said in his paper, the data collection period is critical, and the duration was less than one month. He applied Naïve Bayes and Decision Tree algorithms, but those could not predict numerical values. In addition to that, the accuracies of his study were less than 70%.

Noor and Jan [20] used multiple linear regression to build a car price prediction model in their study. Their dataset collection period was two months. During this time, they collected many features for each sample. However, after manual feature engineering, the only remaining features were price, engine type, and model year. At the end of their project, their model came close to perfection, scoring accuracy of 98%.

A different approach was given by Gongqi and his team [21] as they built a model using Artificial Neural Networks (ANN) rather than linear regression. Previous models could not deal with nonlinear relations between inputs. Hence, they developed a reliable model for dealing with those inputs. As a result, their project was better at predicting car prices than other linear regression models.

3 Methodology

The approach for this study is presented in Diagram 2 below. Data collection is the first step in the building process. This has been done using a Python web scraper with the help of BeautifulSoup and Requests libraries. The program scrapes enough data to train and test the model as the process is automated. After collecting the data, the preprocessing stage is the most important. How well the preprocessing is done can influence the accuracy more than any other stage. Preprocessing tasks involve cleaning missing data, encoding categorical variables as an algorithm needs numerical values, and features scaling. Next, outliers are removed in the feature engineering stage. Data is divided between train and test sets. Each car feature gets its weighting, representing how much it will influence the output. In the last stages, prediction models were generated from various machine learning algorithms and evaluated. To apply the algorithms, TensorFlow and Keras libraries were used. Indicators like MSE, MAE, MAPE, and MSLE were used to choose the best model.

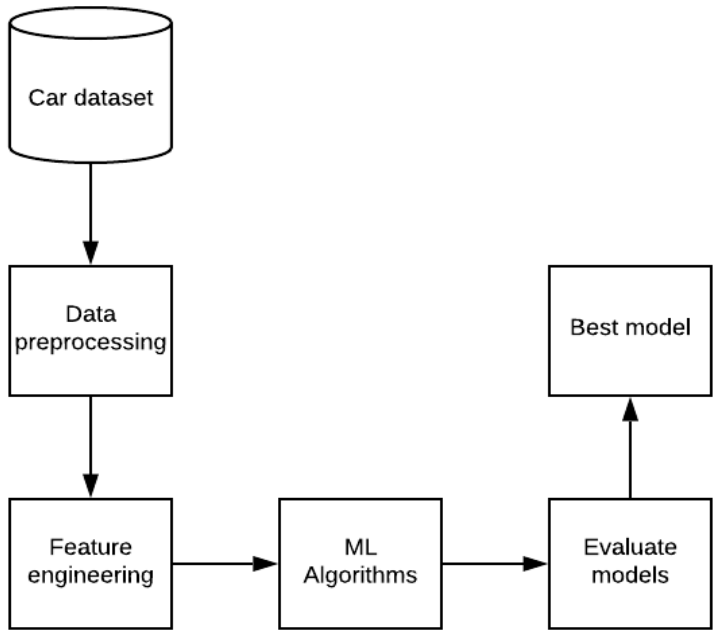

Diagram 2 – Process of building a model

3.1 Software Development Life Cycle

In selecting a Software Development Cycle, a straightforward model was the target. As user interaction is not needed in the study, the Waterfall model is an excellent use choice.

The Waterfall model was the first one introduced and used in software engineering [22]. Each phase of the process depends on the previous one. A phase cannot start if the one before did not finish, like in a linear sequence. Hence, people also refer to it as the "linear-sequential" model. The major drawback of this model is that a team cannot go back to a phase after being stated as finished. The sequential phases of the Waterfall model are [22]:

- Requirement analysis – All the system requirements are gathered and analyzed at this phase. After being analyzed, they are also being documented to be used in the next phase.
- Design – A system is designed based on the specifications gathered at the previous stage. This helps find the hardware requirements and build the overall project's overall architecture.
- Implementation – After the design is ready, each small part of the program is developed and integrated into the testing phase. This prevents building the whole program at once, and debugging will be much easier if problems occur.
- Testing – Each small part developed at the previous stage is tested. After each unit has been tested, the whole program is tested to check for failures.
- Deployment – After the system has been tested, the program is ready to be released.
- Maintenance – Usually, after deployment, errors and issues can appear. In this stage, patches of the initial program are being deployed, and documentation on what has been fixed.

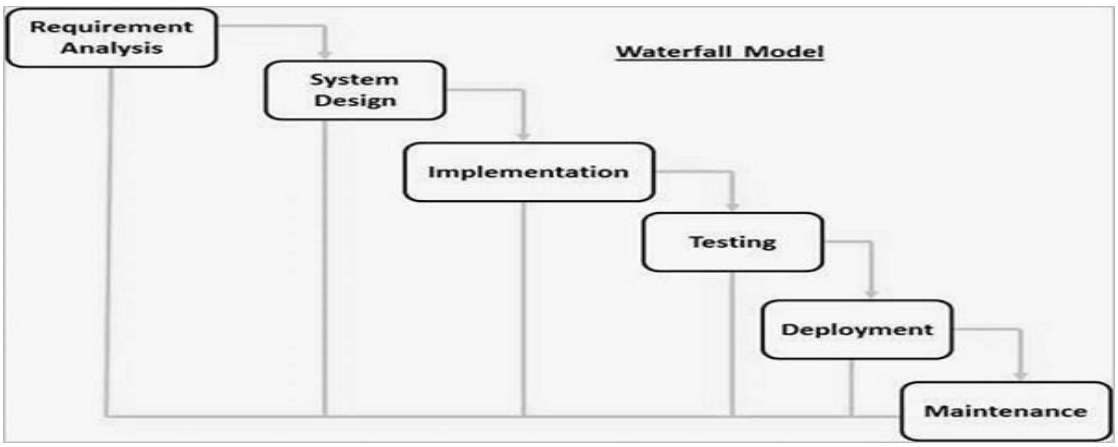

Figure 2 - Waterfall model phases [22]

The main reason we chose Waterfall over other models is that the project is short and straightforward. It does not have ambiguous specifications. If the project had needed user interaction or if the project was more significant, Agile would have been the primary choice. The V-shaped model allows teams to go back to previous stages, but as said previously, it is not needed for this study.

3.2 Tools and Libraries

- **TensorFlow**

TensorFlow [23] is an open-source library for machine learning. It provides fundamental building blocks and functionalities for our neural networks. We used this library to build and train the models.

- **Scikit-learn**

Scikit-learn [24] is a free software machine learning library for Python programming language. It features supervised learning algorithms like support vector machines and random forests. We used this library for a stock implementation of some models.

- **Pandas**

Pandas [25] is a fast, powerful, flexible, and easy-to-use open-source data analysis and manipulation tool built in Python. We used this tool to store the data from the web scraper and then access it.

- **BeautifulSoup**

BeautifulSoup [26] is a Python library for pulling data out of HTML and XML files. We used this library for web scraping the car features.

- **Requests**

Requests [27] is an elegant and simple HTTP library for Python, built for human beings. Requests allow you to send HTTP/1.1 requests exceptionally easily. For example, we used this library to send an HTTP request to scrape the data.

- **Matplotlib**

Matplotlib [28] is a Python library for creating static, animated, and interactive visualizations in Python. We use this library to visualize the data correlation between the features and analyze our training results and the difference between each model's predicted and actual prices.

- **Keras**

Keras [29] is an API designed for human beings, not machines. Nevertheless, it follows the best practices for reducing cognitive load: it minimizes the number of user actions required for everyday use cases and provides clear and actionable error messages. We used this API for creating the CNN for the deep learning model.

3.3 Web Scraping

Manual data collection is tedious and requires a lot of human computation resources, and it is also more error-prone than an automated gathering method. An example of an automated method is a web scraper. This method extracts data from a specific website, and it can store the information locally. A web scraper can mimic human interaction with the website. It can also make requests to get the required information for a project. Another use of a web scraper is that the preprocessing and normalizing stages can be done inside it. An example is when a scraper gets the data displayed in an advertisement title.

A Python web scraper has been developed (Figure 3). The libraries used were BeautifulSoup and Requests. In addition, a Panda DataFrame has been used to store the data. The targeted website is PistonHeads [30]. The program fully scrapes all the features displayed in the car advertisement, only having empty spaces because the data was missing. Using the BeautifulSoup library, three different methods were created. One returns the HTML of the entire web page; this one is used to scrape the information. The code then looks for specific CSS elements of each feature. Another method is for clicking into each car advertisement, as the features presented in the title are not enough. The last method is to go onto the next page with car advertisements, as we want to scrape data from as many cars as possible. After all the features have been extracted and stored in a Panda DataFrame, we do a bit of data preprocessing, for example, wiping out the word “miles” from the “Mileage” column as the model can only work with numerical variables. Finally, the DataFrame is sent into an Excel file for advanced dataset analysis.

```
while True:
    soup = getData(website)
    website = getNextPage(soup)
    nr+=1

    #finds all the cars
    cars = soup.find_all('div', {'class': 'result-contain'})

    for car in cars:

        #car make, car model, manufacture year
        try:
            adTitle = car.find('h3').get_text()#extracting the whole ad title

            age = adTitle[len(adTitle) - 5 :len(adTitle) - 1]
            make = adTitle.split(' ')[0]
            if make == 'Land':#special case when the make of the car is Land Rover
                make = make + ' ' + adTitle.split(' ')[1]

            model = adTitle[len(make) + 1:]
            model = model.split(' (')[0]

            car_make.append(make)
            car_model.append(model)
            car_manufacture_year.append(age)
```

Figure 3 – Name and price of the car are extracted

3.4 Dataset Description

The dataset consists of over 2000 samples and critical features like name, price, mileage, fuel type, gear transmission, and horsepower, alongside extra features that can still influence the model's outcome. The whole set of car features can be seen in Figure 4. The dataset has both numerical values such as price and categorical values like the car make and model.

	Make	Model	Year of manufacture	Body	Doors	Seats	Colour	Engine size	Price	Mileage	Fuel type	Fuel economy	Transmission	Horse power	Owners	Extraction time
2	Seat	Ibiza 1.4 Toca 5dr * REAR SENSORS / LOW MILEAGE / EMOCION RED *	2014	Hatchback	5	5	Red	1.4	£6,995	42,848	Petrol	47.9	Manual	84	2	16-11-2021
3	Jaguar	XE 2L R-Sport d	2018	Saloon	4	5	White	2	£23,025	17,893	n/a	48.7	Automatic	177	1	16-11-2021
4	Ford	Focus 2.0 ST-3 TDCI 5d 183 BHP	2018	Hatchback	5	5	Black	2	£18,500	51,149	n/a	67.3	Manual	182	2	16-11-2021
5	Toyota	AYGO 1.0 VVT-I X-PLAY 5d 69 BHP	2017	Hatchback	5	4	Blue	1	£6,999	33,411	Petrol	68.9	Manual	69	1	16-11-2021
6	Vauxhall	Grandland X 1.2 SPORT NAV S/S 5d 129 BHP	2018	Hatchback	5	5	Blue	1.2	£16,499	42,111	Petrol	55.4	Manual	128	1	16-11-2021
7	Jaguar	F-PACE 2.0 PRESTIGE AWD 5d 178 BHP	2017	Estate	5	5	Blue	2	£28,899	40,264	n/a	53.3	Automatic	177	1	16-11-2021
8	Mercedes	A-Class 2.1 A 200 D AMG LINE PREMIUM 5d 134 BHP	2018	Hatchback	5	5	Grey	2.1	£18,699	58,664	n/a	68.9	Automatic	134	1	16-11-2021
9	Kia	ceed 1.6 CRDI 2 ISG 5d 114 BHP	2018	Hatchback	5	5	Black	1.6	£12,799	28,762	n/a	58.9	Manual	114	1	16-11-2021
10	BMW	3 Series 2.0 330E M SPORT 4d 181 BHP	2018	Saloon	4	5	Grey	2	£21,299	27,723	Hybrid	134.5	Automatic	248	1	16-11-2021

Figure 4 – Custom car dataset

The car features are:

- Make – represents the make of the car. It is a categorical value.
- Model – represents the model of the car. It is a categorical value.
- Year of manufacture – represents the fabrication year. It is a numerical value.
- Body – represents what type the car is. It is a categorical value.
- Doors – represents the number of doors the car has. It is a numerical value.
- Seats – represents the number of seats the car has. It is a numerical value.
- Colour – represents the car's colour. It is a categorical value.
- Engine size – represents the size of the car engine. It is a numerical value.
- Price – represents the car price. It is a numerical value.
- Mileage – represents the number of miles that the car has driven so far. It is a numerical value.
- Fuel type – represents the type of fuel. It is a categorical value.
- Fuel economy – represents the consumption of the car. It is a numerical value.
- Transmission– represents the type of gear. It is a categorical value.
- Horsepower – represents the power of the car. It is a numerical value.
- Owners – represents how many owners it had before it was put up for sale. It is a numerical value.
- Extraction time – represents the time when the web scraping was done. It is a DateTime value.

3.5 Data Preprocessing

It is a crucial stage in the process of building a prediction model. For example, the dataset can contain missing or null values and outliers, causing unwanted noise in the machine learning models. Preprocessing is the method of preparing the data to be used in building a model. Some preprocessing tasks are:

- Checking missing and null values – If there are missing or null values in the dataset, two methods are usually considered. First, the feature is completely removed; this is when many values are missing. Another solution is to introduce the mean deviation of the feature in the missing samples. This is applied when just a few samples are missing.

- Encoding categorical values – Each machine learning algorithm needs numerical values to build a model. A solution to encode the categorical values is to label the feature from 0 to the number of categories–1.

- Features scaling – There is always a difference in the ranges of each feature. To avoid the issue of one feature being predominant, one can normalize the data. The usual technique applied is scaling to a range. This technique creates groups such as 0 – 999 and 1000 – 9999, and each sample will go into its respective group.

3.6 Feature Engineering

It has the same purpose as data preprocessing. The difference is that it involves technical knowledge of machine learning principles. Feature engineering tasks refer to:

- Removing outliers – An outlier is a sample from the dataset with extreme values. An example is when a car has its year of manufacture as 1999 while the others are from 2007 onwards. This creates noise in the algorithm, and it should be removed.

- Feature importance – Each feature has its weight in influencing the prediction. For example, the year of manufacture weighs more than the horsepower of a car. This task helps in reducing overfitting and makes the model faster.

- Dataset splitting – The dataset must be split into a train and test dataset. There are many ways of splitting the data as k-fold Cross-Validation [31] or the 80-20 rule of thumb.

3.7 Implemented Algorithms

The algorithms we implemented are all supervised learning algorithms, as all the features are well known, and that classifies the input data as labelled. Methos used were: Linear Regression, Extreme Gradient Boosting Regressor, Random Forest, and Deep Learning.

- **Linear Regression**

  As the name suggests, Linear regression is a linear model, which a linear relationship between the input data and a single output variable. It can be of two types. If there is only one input variable, it is called simple linear regression. However, if there are multiple input variables, the method is called multiple linear regression [32]. A simple mathematical representation of this method can be as below:

  $$y = B_0 + B_1 * x_1,$$

  where B_0 = bias coefficient and B_1 = coefficient for the height column with y = output variable and x = input variable

  For multiple linear regression, the number of x is equal to the number of inputs and the coefficient are named from B_0 to $B_{\text{number of inputs}}$.

  Since the model parameters are easily interpretable and straightforward, linear regression is one of the most popular models used for regression problems. Moreover, if the dataset is small, it can provide more than satisfactory results [33].

- **Extreme Gradient Boosting**

  Extreme Gradient Boosting, better known as XGBoost, is easy to download and install software library that supports the Python interface, among others programming languages. Tianqi Chen first developed it. It implements a gradient boosting decision tree algorithm at its core, but as [34] suggests, the execution speed and model performance are better than a stock implementation. The key features of its performance are [34]:

  - ✓ Sparse Aware – automatically handling missing data.
  - ✓ Block Structure – supporting the parallelization of tree construction.
  - ✓ Continued Training – boosting an already fitted model to new data.

  Due to its advantages of being both fast and efficient, developers use it in machine learning competitions [35] over other popular solutions like neural networks. It is reliable, but it can also cover a many problems like predictions, classification, motion detection, etc.

- **Random Forest**

  "Random forests are a combination of tree predictors. Each tree depends on the values of a random vector sampled independently and with the same distribution for all trees in the forest" [36]. An advantage of using random forests is that it overcomes the overfitting problems. Usually, if we scale up the method, the overfitting becomes bigger. However, it has been demonstrated in Appendix I of [36] that the more trees we add to a random forest, the chance of overfitting does not go up. However, a limiting value of the generalizing error is produced.

  A drawback of this algorithm is that having a large number of trees will make it too slow and ineffective. The training can still be fast, but creating the predictions classifies this method as unusable for large datasets. However, for most applications, it can still be effective [37].

- **Deep learning**

  The most popular and most used algorithm nowadays. It has the broadest range of applications among all the existing methods. It can be used in speech recognition, object detection, price prediction, and many more fields. Types of deep learning are:

  - ✓ Artificial neural network – used for detection and recognition
  - ✓ Convolutional neural network – used for advanced image recognition

  The main difference between a standard machine learning approach and a deep learning one is that the deep learning model can skip the preprocessing step as it can work with unstructured data. However, the data must be preprocessed to get the best accuracy of a machine learning process [38].

  It had the most influence on computing problems by far, as in the past, a machine learning system demanded high computational resources and considerable domain expertise [39]. Nowadays, everyone can feed their machine with raw data, and it will automatically determine the representations needed for detection or classification. With deep learning, problems where the input could not be converted to an understandable state, could finally be solved.

  Both ANN and CNN use Backpropagation for training. The steps for backpropagation are [40]:

  - Train the data and compute an output
  - Calculate the error between the output and the actual value
  - Adjust the weights
  - If the error is higher than the predefined tolerance, it goes back to step 1; if not, stop.

3.8 Challenges

- **Resources**

As it is an innovative field of study and a developing area, there are not a plethora of theoretical and practical resources about car price prediction. Related works are usually designed to separately check the liability of one method rather than comparing and determining which one is the best. Their system is also more potent than ours at hand, which can influence the outcome of the predictions.

- **Website security**

Our scraper acts like a spider and generates thousands of requests per minute to the website. The system detects it as an uncommon behaviour, and depending on how the website security is designed, the scraper might get blocked and stopped from collecting data. This limits the number of requests the scraper can make, hence the number of resources our dataset can have. If the scraper gets blocked, we must find another website and rewrite the code for the scraper, as every web page has a different design. This has been overcome by limiting the request number the scraper can make to 1000 per day.

- **Training time**

Since our system power is limited, the training time of each deep learning model was often too long, especially when adding lots of extra layers and parameters. Naturally, this affected our running time and ability to test the environment. To mitigate this problem, I used EarlyStopping [41] to stop the program from running if the "val_loss" variable was not improving. Moreover, after seeing that the results of other methods are better than deep learning, we reduced the complexity of our model to a basic one.

- **Overfitting**

As most machine learning projects are sensible to overfitting, ours was not an exception. Training the deep learning model too much would result in overfitting. However, using EarlyStopping [41] made the training faster and avoided overfitting.

- **Hyperparameter sensitivity**

Our experience gathered during this study shows that a deep learning system's performance is susceptible to its hyperparameters, such as learning rate, number of epochs, batch size, kernel size, kernel initializer, etc. Unfortunately, we could only practice hyperparameters empirically from other resources that worked adequately with other learning scenarios.

4 Experiments

4.1 Setup

For our setup, after we collected the data, we added all the car features into a Panda DataFrame, and then we started the preprocessing stage.

Firstly, we deleted all the samples where the car "make" was not found or unavailable as it would not make sense to keep them. Afterwards, we investigated the dataset and tried to see if any features contained letters besides numbers. We found "miles" in the mileage column, "bhp" for horsepower, "+VAT" for the price, and many more. Before going any further, we stripped those letters from their columns as we only need the numerical value of those in building the model. We kept the "£" sign and the comma for price and mileage for esthetical reasons. In the number of more than 1200, all the samples have been moved to a .csv file at the end of the web scraper part.

```
#deleting data where the car name
# could have not been scraped
data = data.loc[data['Make'] != 'n/a']

# Data cleaning
# I want only numbers in the Excel file
data ['Mileage'] = data['Mileage'].apply(lambda x: x.strip('miles'))
data ['Horse power'] = data['Horse power'].apply(lambda x: x.strip('bhp'))
data ['Price'] = data['Price'].apply(lambda x: x.strip('+VAT'))
data ['Engine size'] = data['Engine size'].apply(lambda x: x.strip('L'))
data ['Fuel economy'] = data['Fuel economy'].apply(lambda x: x.strip('mpg'))
```

Figure 5 - Preprocessing data in the web scraper

After the dataset had been loaded in the model building file, we checked how many samples had missing features from the scraping part and printed their number with that feature missing. We decided to drop all the columns with more than 100 samples missing that feature. The columns were "Fuel type," "Doors," and "Body." Furthermore, "Extraction time," "Model," "Seats," and "Colour" were also dropped because we decided that those do not influence the model accuracy as much as the other features.

```
data.drop('Fuel type', inplace=True, axis=1)
data.drop('Doors', inplace=True, axis=1)
data.drop('Seats', inplace=True, axis=1)
data.drop('Body', inplace=True, axis=1)
data.drop('Extraction time', inplace=True, axis=1)
data.drop('Model', inplace=True, axis=1)
data.drop('Colour', inplace=True, axis=1)
```

Figure 6 - Removing unimportant columns

After the columns were successfully dropped, we checked again for samples with missing features. We found some with “Owners,” “Horse power,” “Fuel economy,” “Engine size,” and “Mileage.” Instead of removing the entire column, we settled on removing only the faulty samples.

```
data = data.loc[data['Owners'] != 0]
data = data.loc[data['Fuel economy'] != 0]
data = data.loc[data['Engine size'] != 0]
```

Figure 7 - Removing samples with missing features

Later, we converted the” Price” and “Mileage” columns to a numerical format and removed the comma for both and also the “£” sign for “Price.” We did it because a correlation between a categorical and numerical value cannot be done.

```
data["Price"] = data["Price"].str.replace(",", "")
data['Price'] = data['Price'].apply(lambda x: x.strip('£'))
data['Price'] = pd.to_numeric(data['Price'])
data["Mileage"] = data["Mileage"].str.replace(",", "")
data['Mileage'] = pd.to_numeric(data['Mileage'])
```

Figure 8 - Converting Price and Mileage to numerical format

As stated above, correlations between categorical and numerical features do not exist. Therefore, we had to encode our categorical variables. Those were “Make” and “Transmission.”

```
labelencoder = LabelEncoder()
data["Make_N"] = labelencoder.fit_transform(data["Make"])
data["Transmission_N"] = labelencoder.fit_transform(data["Transmission"])
```

Figure 9 - Encoding the categorical features

The last significant step in preprocessing the data is to remove the outliers, which can influence the accuracy of our models. Before removing, we need to analyse if there are outliers in our dataset. Those are outside the interval in the following plots.

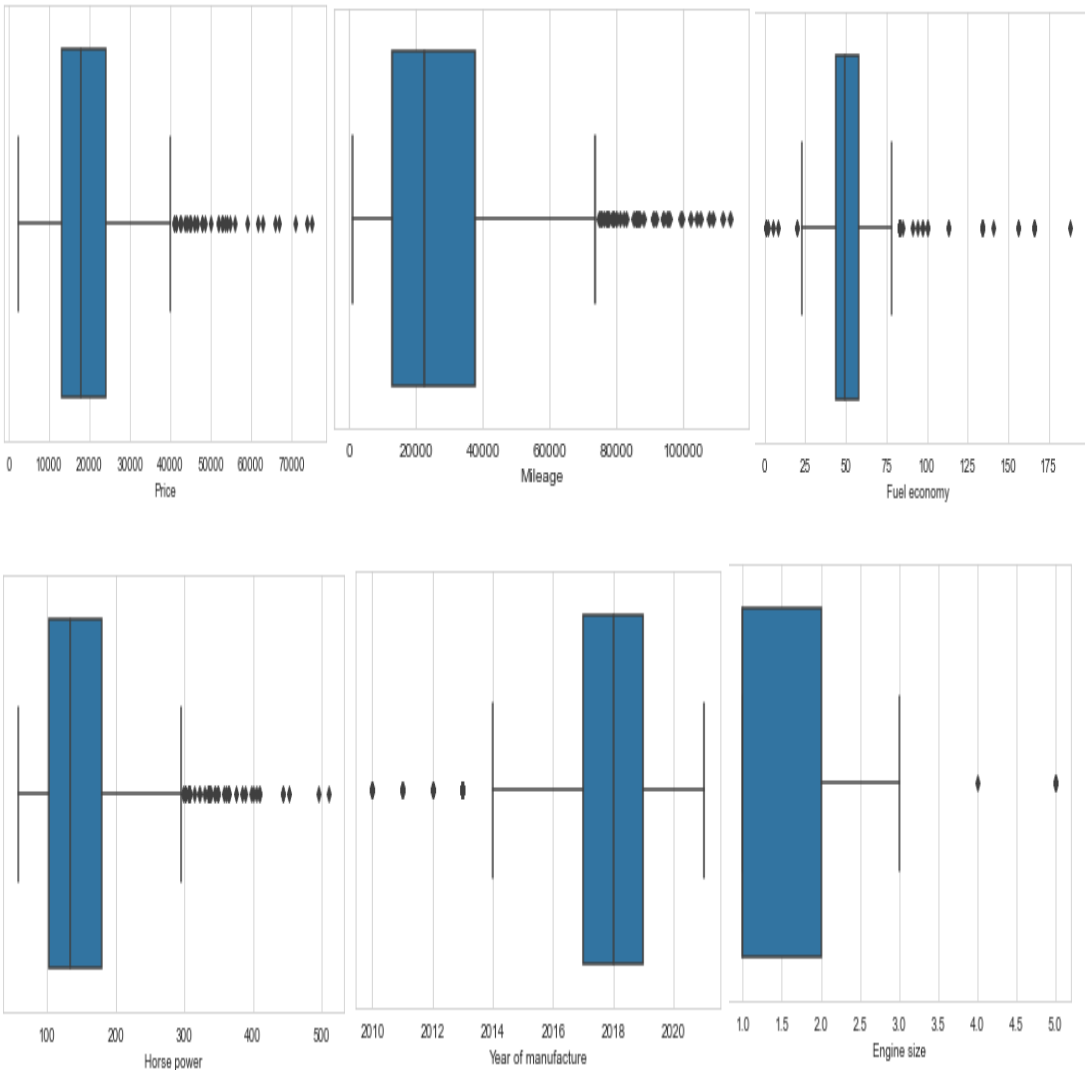

Figure 10 -Plots of each feature

As seen in the plots above, all the remaining features have outliers. Therefore, we need to remove all the respective samples.

```
outlier_cols=['Price','Mileage', 'Fuel economy','Horse power','Year of manufacture', 'Engine size',
'Owners']
for col in outlier_cols:
    lower,upper=find_outliers_limit(data,col)
    data[col]=remove_outlier(data,col,upper,lower)
```

Figure 11 – Removing the outliers from the dataset

Before splitting and scaling the data for training and testing, we displayed the correlations between each feature and tried to find suitable weightings to get the best accuracy.

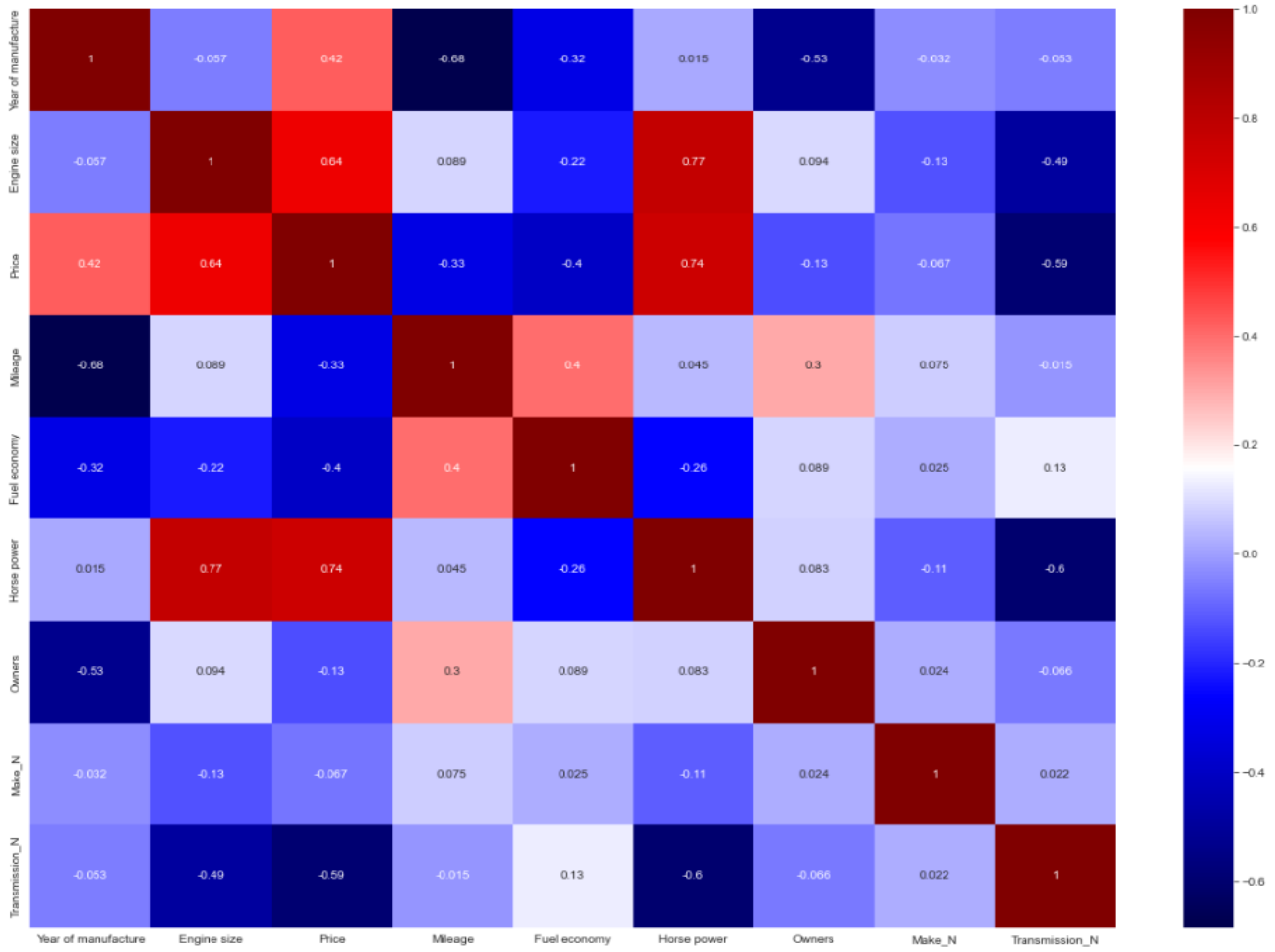

Figure 12 - Data correlation between each feature

4.2 Results

4.2.1 Linear Regression

As seen in the figure below, linear regression does an overall good job of predicting the prices. The main problems of this model are when predicting the price for expensive cars (see observations 37 and 63) and when there is a substantial difference between the prices of the following cars in the dataset. Compared to the other models, it is worse than XGBoost and random forest but better than deep learning.

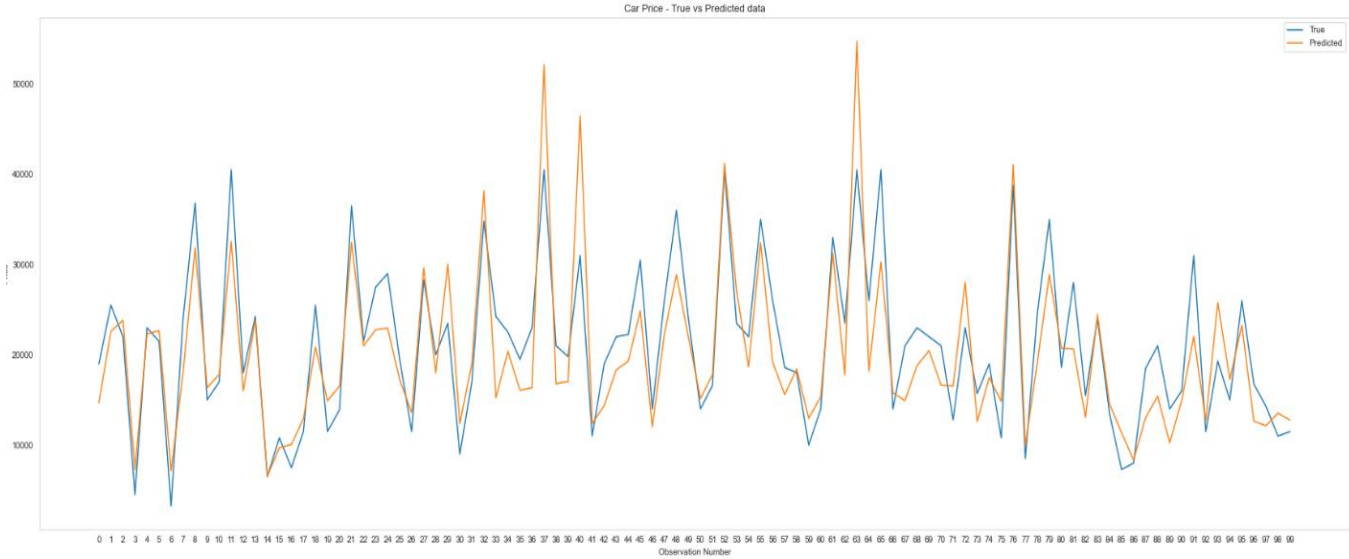

Figure 13 - Linear Regression price graph compared to the actual price

4.2.2 Extreme Gradient Boosting

It has no visible issues with the predicted prices, the only significant difference being in observation 53. It is twice as good as linear regression and deep learning and equal to a random forest, but its main advantage is the speed at which the model is built and trained. It is the quickest method of the ones we tested in this study.

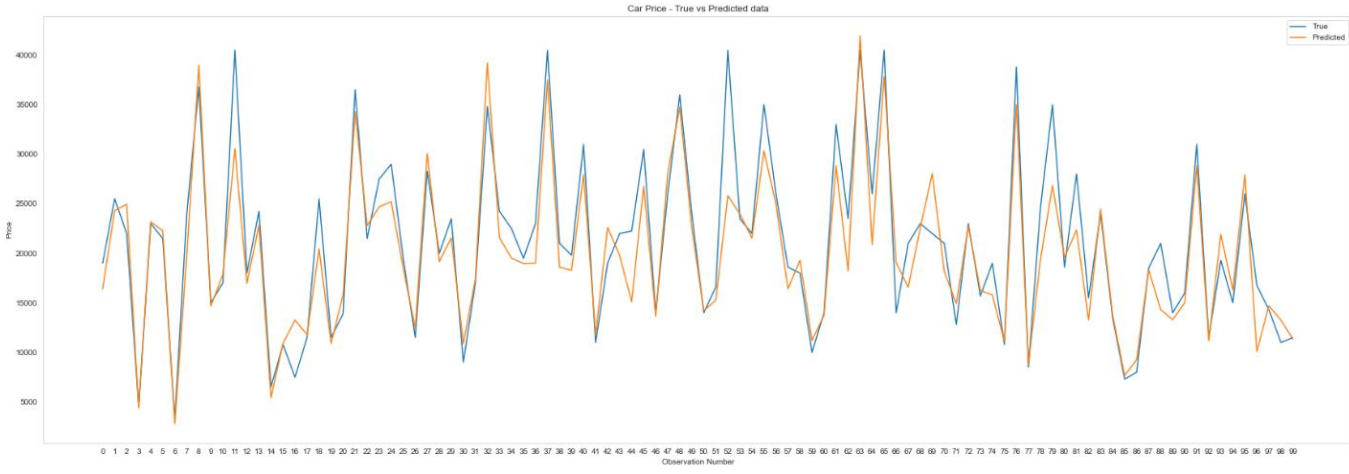

Figure 14 - XGBoost price graph compared to the actual price

4.2.3 Random Forest

It qualifies as a reliable model because its prediction is better than linear regression and deep learning and also equal to XGBoost.

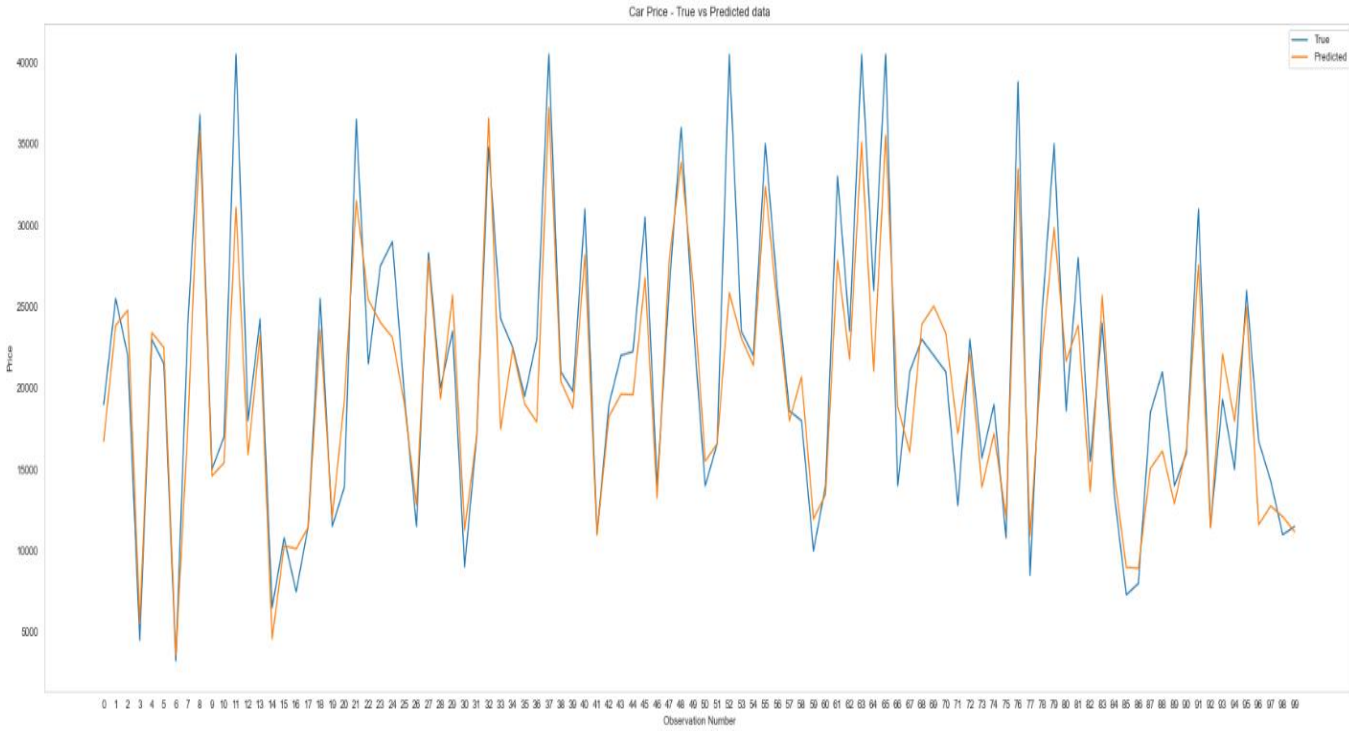

Figure 15 - Random Forest price graph compared to the actual price

4.2.4 Deep Learning

The model uses three dense layers of 128, 64, and 32 dimensions and one output layer of dimension 1. The input dimension of the layer is the number of features that remained – 1 because we leave out the price column. The activation function used was Rectified Linear Unit (ReLU) and Linear for the output layer. The loss value of the model was measured using "mean_squared_error," and the optimizer was "adam." The number of epochs for the model was 100, with the batch size 10. In addition to a standard implementation of deep learning, we used an EarlyStopping [41] variable. This variable would make the program run faster and more efficiently as it monitors the val_loss of the model, and it will stop the training if the variable does not decrease after three epochs. A good observation about this approach is that the loss and validation loss values are below 1, meaning our implementation does not overfit.

In terms of accuracy, it is the worst model we implemented, and it also uses many computation resources. So, we decided not to build a complicated and demanding model as it would be slow, and the accuracy does not improve to be worth it. This applies to the small dataset that we gathered.

However, for large datasets, deep learning would probably be the most efficient model as then it will have lots of samples for training, which would significantly improve the accuracy. Also, a more powerful machine would be a significant advantage for using deep learning.

Figure 16 - Loss and validation loss of the deep learning model

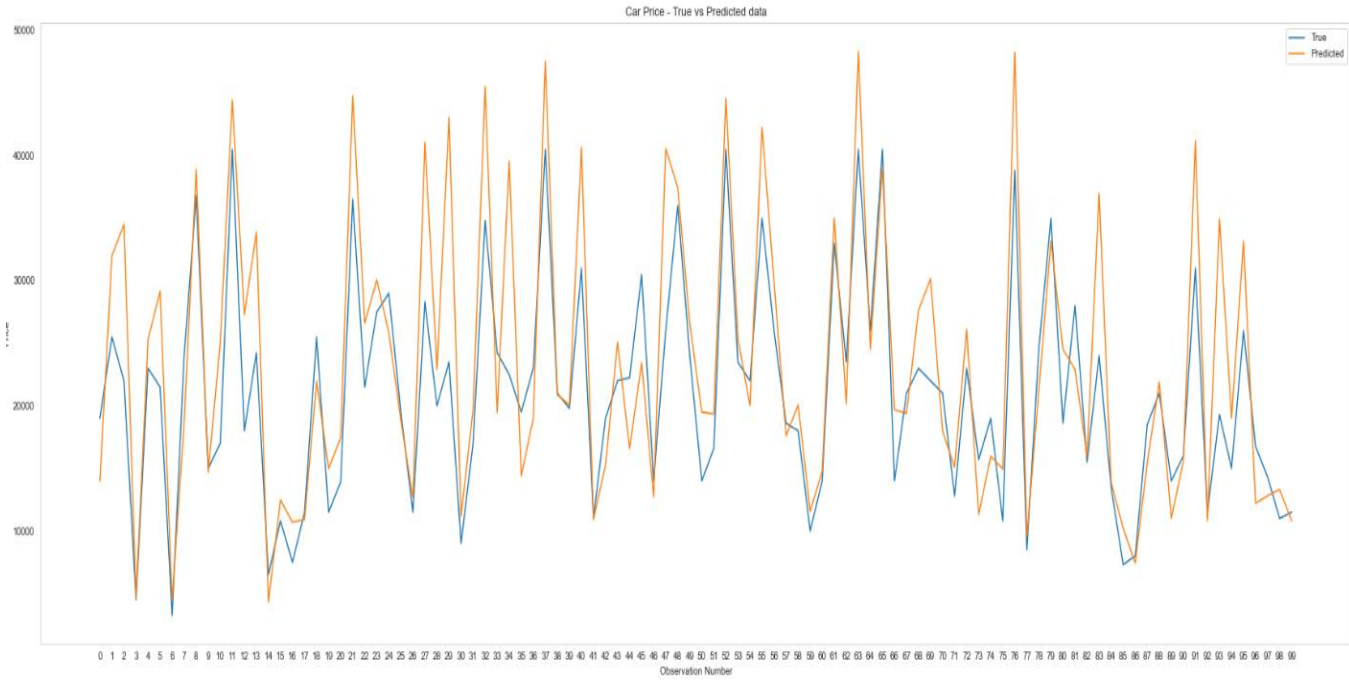

Figure 17 - Deep learning price graph compared to the actual price

4.2.5 Best model

To compare the results of each model, we have used the following indicators:

- Mean squared error (MSE) – "tells you how close a regression line is to a set of points. It does this by taking the distances from the points to the regression line (these distances are the "errors") and squaring them. The squaring is necessary to remove any negative signs" [42]

- Mean absolute error (MAE) – "measures the average magnitude of the errors in a set of forecasts, without considering their direction" [43]

- Mean absolute percentage error (MAPE) – "is a measure of how accurate a forecast system is. It measures this accuracy as a percentage and can be calculated as the average absolute percent error for each period minus actual values divided by actual values" [44]

- Mean squared logarithmic error (MSLE) – "can be interpreted as a measure of the ratio between the true and predicted values"[45]

Model	MSE	MAE	MAPE	MSLE
LR	25118657.59	3684.09	0.2	0.06
XGB	10138474.81	2280.11	0.12	0.03
RF	9790039.58	2371.8	0.13	0.03
DL_LARGE	32838868.77	4207.41	0.22	0.06

Figure 18 - Model results

As the table above shows, the best model we implemented was extreme gradient boosting. Apart from MSE, it had the lowest values for the other indicators. In terms of speed, XGBoost is also the quickest model that trains and builds the model for our dataset.

5 Project Management

We achieved everything proposed at the beginning of this project in terms of implementation. Furthermore, we successfully managed our time to fully develop all the implementation parts starting from creating the scraper to choosing the best model suitable for predicting the car prices.

For the Initial Document deadline, the implementation that needed to be ready at that time was both developed and tested. Hence, the amount of information presented in the document was filled with initial findings and discussions about the project.

For Gregynog, we presented our initial results for the first algorithms tested, neural networks, and support vector machines. At that time, the progress was on track to be finished before the final deadline.

As mentioned in Section 5.1, the program was entirely implemented for the dissertation deadline, and the final document was thoroughly written.

5.1 Schedule

As presented in Appendix A, the project was developed with the initial work schedule.

When writing the initial document, the project was at the first milestone – the web scraper was developed, and the dataset was collected and created. After that, the selection of machine learning algorithms started on time. In that milestone, the selected algorithms were trained and tested on the custom dataset collected. Before Gregynog, all the desired algorithms were tested. After the presentation, hyperparameter tweaking was done to improve the accuracy of the models. Finally, we chose the best model based on the last evaluation metrics we obtained in the final implementation phase.

The first milestone of our project – creating a web scraper and collecting data – has been completed successfully. The scraper collected data while not detected as an uncommon user or behaviour.

The second milestone – selecting, training, and testing machine learning algorithms – has also been concluded. We tried various algorithms like k-nearest neighbours, support vector machines, and multiple linear regression, but in the end, we kept the algorithms already presented in Chapter 3.

The last milestone – choosing the best model – has been done by looking at the table from Section 4.2.5 and extracting the model with the lowest errors.

Finally, we can say that we achieved all the milestones that we wanted in the Initial Document. However, more tweaking for the deep learning model would have been more satisfying.

5.2 Risks

The risks associated with the project were analysed in detail in the initial document, and they are presented in Appendix B. Most mitigation strategies prevented the predicted risks from occurring. However, the following risks had an impact on the project:

- Infection with Covid-19 – during the infection period, the feeling of dizziness made daily activities hard to achieve. Hence, the progress of the study was slow over the winter.

- Being banned from websites – in collecting data, the scraper was being detected as an uncommon behaviour, which resulted in IP being banned from accessing the website. However, a VPN tricked the system, and afterwards, the data collection went with no fuss.

- Loss of data – data was already collected before the scraper got banned, which erased all the samples collected, and a new dataset had to be created.

6 Future Work and Conclusion

6.1 Reflection on Results

We successfully developed a web scraper to collect the data in this project. Then we preprocessed all the samples before training various machine learning models on the custom dataset.

Furthermore, we compared different algorithms using our environment through all the experiments on different models and approaches. Finally, putting the results of the experiments in context with each other, we set out to explore that deep learning models can suffer severely from hyperparameter tweaking and its primary construction. It also needs a large dataset for the training to be compelling and show its full potential in predicting prices.

Our results also showed that XGBoost implementation could be very effective and fast. It can be a good alternative for studies that do not have powerful computational resources. It is easy to use and presents as a library for Python.

6.2 Reflection on Project

Our initial objectives for this project were:

- To explore the machine learning subject and understand how it works
- To identify the web scraping challenges and risks
- To build a car price prediction model which has a high accuracy
- To produce specification and evaluation criteria

We successfully achieved all the main goals while we could not accomplish an even better accuracy due to the limitations in time and scope. The system's complexity has been thoroughly analysed at the beginning of the project, and it has been proved to be a correct and sincere analysis as the program is as complex as we thought it would be. Training time became a significant challenge during our project development phase, as we tested advanced neural networks. To mitigate this problem, and because the accuracy was not higher than other approaches, we reduced the complexity of the neural network while trying to optimise the training pipeline.
In producing specification and evaluation metrics, we successfully compared all the results from different approaches and created a table where we extracted the best model.

6.3 Future Work

- **Hyperparameter search**

The algorithms we adopted all use hyperparameters that are empirically shown to be suitable for other environments. Unfortunately, we only had the opportunity to tweak specific hyperparameters to obtain better performances moderately. In the future, with ample time, a potential run of hyperparameter search could further boost the performances of our solutions.

- **Complex datasets**

Our approach to deep learning had visible problems working with a small dataset. In general, for a complex method like neural networks, the input data must be very generous to use its potential. Again, with more time, finding a better website for scraping data, one that has more resources, would probably improve the accuracy of our deep learning model.

- **Inspecting features correlations**

Although we displayed a data correlation plot for the features, we did not go deeply into it. For example, we did not inspect if a remaining feature can influence the accuracy more than the rest. It would be desirable to find the whole correlations and adjust the weighting of each feature for the deep learning model.

- **Powerful resources**

As mentioned before, a robust approach like neural networks also requires strong machines in terms of computational resources. Even though we did not have access to a robust machine, the results were not influenced much by that as the dataset was small. For a larger dataset, a powerful machine could make a big difference.

6.4 Summary

In this project, we designed and implemented a web scraper, collected data, and implemented Supervised Learning algorithms, including Linear Regression, Extreme Gradient Boosting, Random Forest, and Deep Learning. We compared these algorithms and solutions by providing a specification and evaluation metrics. We could visualize the training process and results, expressing performance indicators like MSE, MAE, MAPE, and MSLE. As a result, we discovered that XGBoost was the most accurate, efficient, and fast approach implemented within our environment. Furthermore, we were able to observe exciting model behaviours.

7 References

[1]" Number of Cars in the UK 2021", NimbleFins, last updated October 4, 2021, https://www.nimblefins.co.uk/cheap-car-insurance/number-cars-great-britain.
[2]" The Ultimate List of UK Car Stats 2020", CarMoney, last updated June 4, 2020, https://www.carmoney.co.uk/blog/the-ultimate-list-of-uk-car-stats.
[3]" How Popular is Car Leasing in the UK now? (2021)", Rivervale, last updated July 29, 2021, https://www.rivervaleleasing.co.uk/blog/posts/how-popular-is-car-leasing.
[4] Mariana Listiani, "Support Vector Regression Analysis for Price Prediction in a Car Leasing Application "(MSc thesis, Hamburg University of Technology, 2009).
[5]" Total number of new passenger cars registered in the United Kingdom (UK) from 2003 to 2020", Statista, last updated August 12, 2021,https://www.statista.com/statistics/299240/volume-of-new-passenger-cars-registered-in-the-united-kingdom/.
[6]" Technology forecasting", Wikipedia, last updated September 27, 2021, https://en.wikipedia.org/wiki/Technology_forecasting.
[7]" Machine learning", Wikipedia, last updated October 22, 2021, https://en.wikipedia.org/wiki/Machine_learning.
[8]" Companies Will Spend $50 Billion On Artificial Intelligence This Year With Little To Show For It, Forbes, last updated October 20, 2020, https://www.forbes.com/sites/davidjeans/2020/10/20/bcg-mit-report-shows-companies-will-spend-50-billion-on-artificial-intelligence-with-few-results/?sh=781e3c0d7c87.
[9] Ethem Alpaydin, *Introduction to Machine Learning* (Massachusetts: MIT Press, 2004).
[10] Kevin Patrick Murphy, *Machine Learning: A Probabilistic Perspective* (Massachusetts: MIT Press, 2012).
[11]" Overfitting and Underfitting With Machine Learning Algorithms", Machine Learning Mastery, last updated August 2, 2019, https://machinelearningmastery.com/overfitting-and-underfitting-with-machine-learning-algorithms.
[12]"What is underfitting and overfitting in machine learning and how to deal with it.", Medium, last updated March 11, 2018, https://medium.com/greyatom/what-is-underfitting-and-overfitting-in-machine-learning-and-how-to-deal-with-it-6803a989c76.
[13] Chunyang Su et al., "Some progress of supervised learning," in *Advanced Intelligent Computing Theories and Applications*, ed. De-Shuang Huang et al. (Shanghai: 4th International Conference on Intelligent Computing, 2008)
[14]" 6 Types of Supervised Learning You Must Know About in 2021", upGrad, last updated January 10, 2021, https://www.upgrad.com/blog/types-of-supervised-learning.

[15]" Artificial Neural Network (ANN)", techopedia, last updated June 21, 2021, https://www.techopedia.com/definition/5967/artificial-neural-network-ann.
[16] Zabir Haider Khan et al., "Price Prediction of Share Market using Artificial Neural Network (ANN), *International Journal of Computer Applications* 22, no.2(2011): Article 8.
[17]" Artificial Neural Networks Advantages and Disadvantages" LinkedIn, accessed October 26, 2021, https://www.linkedin.com/pulse/artificial-neural-networks-advantages-disadvantages-maad-m-mijwel/.
[18] Michael S. Richardson, "Determinants of used car resale value" (MSc thesis, Colorado College, 2009).
[19] Sameerchand Pudaruth, "Predicting the price of used cars using machine learning techniques," *International Journal of Information and Computation Technology* 4, no. 7(2014):753-764.
[20]Noor, K., & Jan, S. (2017). Vehicle price prediction system using machine learning techniques. *International Journal of Computer Applications*, *167*(9), 27–31., https://doi.org/10.5120/ijca2017914373.
[21] Shen Gongqi, Wang Yansong, & Zhu Qiang. (2011).
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*. https://doi.org/10.1109/icmtma.2011.455
[22]" SDLC – Waterfall Model", tutorialspoint, accessed October 25, 2021, https://www.tutorialspoint.com/sdlc/sdlc_waterfall_model.htm.
[23]" An end-to-end open-source machine learning platform", TensorFlow, accessed April 20, 2022, https://www.tensorflow.org/.
[24]" Machine Learning in Python", scikit-learn, accessed April 20, 2022, https://scikit-learn.org/stable/.
[25]" Pandas", pandas, accessed April 20, 2022, https://pandas.pydata.org/
[26]" Beautiful Soup Documentation", crummy, accessed April 20, 2022, https://www.crummy.com/software/BeautifulSoup/bs4/doc/.
[27]" Requests: HTTP for Humans", docs.python, accessed April 20, 2022, https://docs.python-requests.org/en/latest/.
[28]" Matplotlib: Visualization with Python", matplotlib, accessed April 20, 2022, https://matplotlib.org/.
[29]" Keras", keras, accessed April 20, 2022, https://keras.io/ .
[30]" Piston Heads", PistonHeads, accessed October 25, 2021, https://www.pistonheads.com/.
[31]" A Gentle Introduction to k-fold Cross-Validation", Machine Learning Mastery, last updated August 3, 2021, https://machinelearningmastery.com/k-fold-cross-validation/.
[32]" Linear Regression for Machine Learning", Machine Learning Mastery, last updated August 15, 2020, https://machinelearningmastery.com/linear-regression-for-machine-learning/
[33] Su, X., Yan, X., & Tsai, C.-L. (2012).
Linear regression. *Wiley Interdisciplinary Reviews: Computational Statistics*, *4*(3), 275–294. https://doi.org/10.1002/wics.1198

[34]" A gentle introduction to XGBoost for Applied Machine Learning", last updated February 17, 2021, https://machinelearningmastery.com/gentle-introduction-xgboost-applied-machine-learning/.
[35] Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*., https://doi.org/10.1145/2939672.2939785.
[36] L. Breiman, Random forests. *Machine Learning*, 45(1):5–32, Oct. 2001.
[37]" Random Forest Algorithm: A Complete Guide", builtin, last updated April 14, 2022, https://builtin.com/data-science/random-forest-algorithm .
[38]" Deep Learning", ibm, accessed April 20, 2022, https://www.ibm.com/cloud/learn/deep-learning .
[39] LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*(7553), 436–444. https://doi.org/10.1038/nature14539.
[40] Zabir Haider Khan et al., "Price Prediction of Share Market using Artificial Neural Network (ANN), *International Journal of Computer Applications* 22, no.2(2011): Article 8.
[41]" EarlyStopping", tensorflow, accessed April 20, 2022, https://www.tensorflow.org/api_docs/python/tf/keras/callbacks/EarlyStopping .
[42]" Mean Squared Error: Definition and Example", statisticshowto, accessed April 20, 2022, https://www.statisticshowto.com/probability-and-statistics/statistics-definitions/mean-squared-error/ .
[43]" Mean Absolute Error(MAE) and Root Mean Squared Error(RMSE)", eumetrain, accessed April 20, 2022, http://www.eumetrain.org/data/4/451/english/msg/ver_cont_var/uos3/uos3_ko1.htm .
[44]" Mean Absolute Percentage Error (MAPE)", statisticshowto, accessed April 20, 2022, https://www.statisticshowto.com/mean-absolute-percentage-error-mape/ .
[45]" Mean squared logarithmic error (MSLE)", peltarion, accessed April 20, 2022, https://peltarion.com/knowledge-center/documentation/modeling-view/build-an-ai-model/loss-functions/mean-squared-logarithmic-error-(msle) .

Appendix A

The Work Schedule has been detailed in the Initial Document. Figure A.1 presents the planned schedule in the form of a Gantt chart.

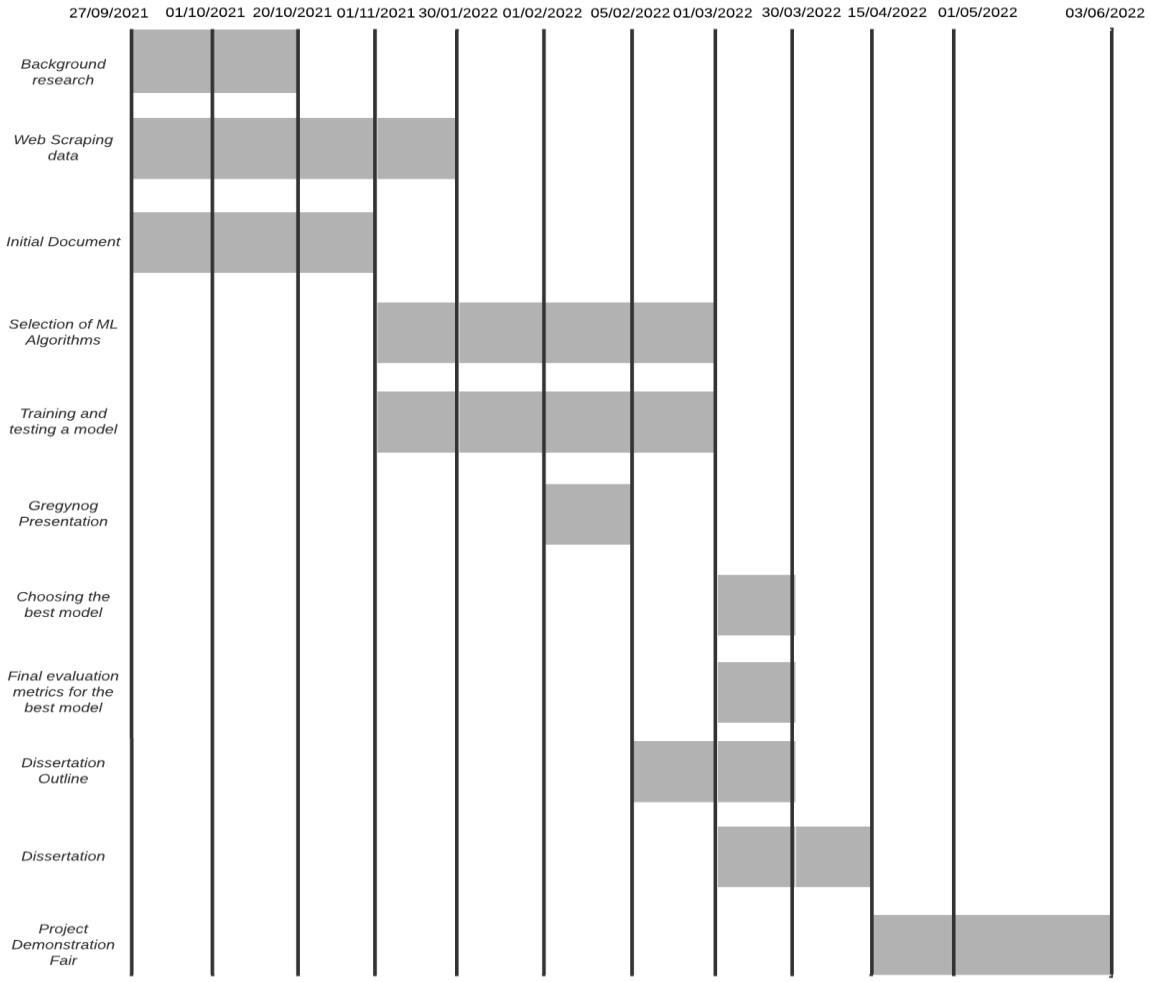

Figure A.1: Work Schedule in the form of a Gantt chart

Appendix B

Risk	How likely?	How may it endanger the project?	How can it be avoided?
Infection with Covid-19	1/5	In the recovery period, I might not be able to continue the project.	I will be wearing face masks when going outside, wash my hands regularly, and, as much as possible, avoid face-to-face contact.
Poor work ethic	1/5	Medical issues like eye strain, strain injury, and back pain can appear. Those will affect my condition and my capability to resume the project.	After long periods of study, I will take short breaks and try as much as possible to have a good posture when sitting down and good lighting in my room.
Loss of data	3/5	Data saved on the device could be lost. This would mean that I must scrape data again. The period when the data is scraped is essential so that evaluation metrics differ from the initial ones.	Saving data to external sources like Google Drive and USB sticks can prevent data loss. Moreover, I will have an antivirus installed, which will prevent cyber-attacks.
Being banned from websites	2/5	Websites will treat the web scraper as a robot. If the number of requests is higher than the website allows, my computer will be IP blocked. This will limit the number of resources I can get.	Scraping from websites known for allowing data to be scraped from them will lower the probability of getting banned. In addition, creating proxies and rotating through them while scraping can lower the chance of getting blocked even more.
Overfitting	4/5	If overfitting occurs, the metrics will be greater than the model's performance.	In the data preprocessing stage, I will shuffle the data so a pattern cannot be made in the training set.

Figure B.1: Risk analysis of the project