CZECH UNIVERSITY OF LIFE SCIENCES PRAGUE

Faculty of Economics and Management

Department of Information Engineering

DIPLOMA THESIS

ASSESSMENT OF SUPERVISED LEARNING TECHNIQUES: CASE OF BIKESHARING

Bc. Anastassiya Li

CZECH UNIVERSITY OF LIFE SCIENCES PRAGUE

Faculty of Economics and Management

DIPLOMA THESIS ASSIGNMENT

Bc. Anastassiya Li

Systems Engineering and Informatics
Informatics

Thesis title

Assessment of supervised learning techniques: case of bikesharing

---

Objectives of thesis

The goal of this thesis is using regression and neural network models estimate demand of bikesharing. It will include: Conduct regression analysis, Training of neural network, Validation of findings from both approaches, Determining the accuracy and robustness of results.

Methodology

In the literature review, there will be a study of professional and scientific sources. Two techniques of supervised learning will be characterised. In the practical part, there will be data preparation and analysis model development and evaluation.

The proposed extent of the thesis

60 – 80 pages

Keywords

neural network, training, testing, regression, MSE, R, bikesharing

---

Recommended information sources

BATAINEH, Mohammad; MARLER, Timothy. Neural network for regression problems with reduced training sets. Neural networks, 2017, 95: 1-9.

KUHN, Max, et al. Applied predictive modeling. New York: Springer, 2013.

PENG, Roger. Exploratory data analysis with R. Lulu. com, 2012.

Expected date of thesis defence

2022/23 SS – FEM

The Diploma Thesis Supervisor

doc. Ing. Vojtěch Merunka, Ph.D.

Supervising department

Department of Information Engineering

Electronic approval: 20. 3. 2023

Ing. Martin Pelikán, Ph.D.

Head of department

Electronic approval: 28. 3. 2023

doc. Ing. Tomáš Šubrt, Ph.D.

Dean

Prague on 28. 03. 2023

DECLARATION

I declare that I have worked on my master's thesis titled "Assessment of supervised learning techniques: case of bikesharing" by myself and I have used only the sources mentioned at the end of the thesis. As the author of the master's thesis, I declare that the thesis does not break any copyrights.

Prague March 31, 2023

. . . .

Bc. Anastassiya Li

ACKNOWLEDGEMENT

I would like to thank my family and friends for their support, patience and encouragement during my studies and work on this thesis.

ABSTRACT

The topic of forecasting is relevant, as obtaining the most accurate prediction of the future allows making the most advantageous decisions for achieving positive results in any sphere of human activity, including social. Research subject of this study deals with prediction of bikesharing hourly count using regression analysis and neural network. R is chosen as a programming language for realisation. The first section of this paper provides the literature review needed to understand the topic of the work and the methods used. The second section focuses on practical implementation on selected techniques. The third section describes results of predictions and their performance evaluation. The relevance of the thesis is due to forecasting demand in general and supervised learning techniques assessment.

Key words: neural network, training, testing, regression, MSE, R, bikesharing.

ABSTRAKT

Téma prognózy je relevantní, neboť získání nejpřesnější předpovědi budoucnosti umožňuje učinit nejvýhodnější rozhodnutí pro dosažení pozitivních výsledků v jakékoli oblasti lidské činnosti, včetně sociální. Výzkumný předmět této studie se zabývá predikcí hodinového počtu bikesharingu pomocí regresní analýzy a neuronové sítě. R je vybrán jako programovací jazyk pro realizaci. První část tohoto článku poskytuje přehled literatury potřebný k pochopení tématu práce a použitých metod. Druhá část se zaměřuje na praktickou implementaci vybraných technik. Třetí část popisuje výsledky předpovědí a jejich hodnocení výkonnosti. Relevance práce je dána prognózováním poptávky obecně a hodnocením technik učení s učitelem.

Klíčová slova: neuronová síť, trénování, testování, regrese, MSE, R, bikesharing.

Contents

1 Introduction **1**

2 Objectives and Methodology **2**
2.1 Objectives 2
2.2 Methodology 2

3 Literature Review **3**
3.1 Bikesharing 3
3.1.1 History of development 3
3.1.2 Overview of impact 7
3.2 Supervised machine learning 8
3.2.1 Machine learning process 9
3.2.2 Applications 11
3.3 Types of problems 12
3.3.1 Classification problem 13
3.3.2 Regression problem 14
3.4 Machine learning techniques 15
3.4.1 Regression analysis 15
3.4.2 Neural networks 17
3.4.3 Other techniques 20
3.5 Performance assessment 22
3.5.1 Mean Absolute Error 22
3.5.2 Mean Squared Error 23
3.5.3 Root Mean Square Error 23

4 Practical Part **24**
4.1 Dataset description 24
4.1.1 Data preprocessing 25
4.2 Data Analysis 26
4.2.1 Numerical summary 26
4.2.2 Graphical analysis 27
4.2.3 Task specification 33
4.2.4 Correlation analysis 34
4.3 Regression model 35
4.3.1 Train-Test Split 35
4.3.2 Training 35
4.3.3 Prediction 37

4.4 Neural network model . . . 38
4.4.1 Normalisation . . . 38
4.4.2 Train-Test Split . . . 38
4.4.3 Training . . . 38
4.4.4 Prediction . . . 40

5 Results and Discussion **42**

6 Conclusion **45**

Bibliography **46**

List of Figures

1 Bikesharing generations. Source: Own processing according to (DEMAIO, 2009) and (SHAHEEN, 2014) . . . 3
2 World map of bikesharing. Source: (DEMAIO *et al.*, 2023) . . . 6
3 Model training. Source: (Google Developers, 2022) . . . 8
4 Applications of supervised learning. Source: Own processing according to (MORDENSKY *et al.*, 2022) . . . 12
5 Example of classification task. Source: Own processing using Edgar Anderson's Iris Data (ANDERSON, 1935) . . . 13
6 Example of regression task. Source: Own processing using Longley's Economic Regression Data (LONGLEY, 1967) . . . 14
7 Neural network architecture. Source: (VENKATESWARLU, 2022) . . . 18
8 Example of decision tree. Source: (RANI *et al.*, 2022) . . . 20
9 Illustration of support vector machine. Source: (RANI *et al.*, 2022) . . . 22
10 Output from head function . . . 25
11 Structure of raw data . . . 25
12 Structure of adjusted data . . . 26
13 Summary statistics . . . 27
14 Variances . . . 27
15 Standard deviations . . . 27
16 Histograms . . . 28
17 Boxplots . . . 29
18 Scatter plots for season, year, holiday, working day, weather situation . 31
19 Scatter plots for month, hour, weekday . . . 32
20 Scatter plots for temperature, feeling temperature, humidity, wind speed 33

21 Correlogram . . . 34
22 Mean and variance of count . . . 35
23 Overdispersion test for count . . . 35
24 Regression model . . . 36
25 Regression model - predicted vs actual count . . . 37
26 Neural network visualisation . . . 39
27 Resulting matrix of neural network model . . . 40
28 Neural network model - predicted vs actual count . . . 41
29 Table of MAE values . . . 42
30 Table of MSE values . . . 42
31 Table of RMSE values . . . 43
32 Excerpt from neural network training . . . 44

1 Introduction

> Every time I see an adult on a bicycle, I no longer despair for the future of the human race.
>
> *The Wheels of Chance*
> Herbert George Wells

Bikesharing is a service where people can rent bikes that are present for short-term public use at a price or free of charge. Specialists estimate that more than 3,000 cities (O'SULLIVAN, 2022) worldwide are participating in such program. These systems generally intended to reduce traffic jams, buzz and improve air quality by offering low-cost access to bikes over short-range rides in metropolitan areas unlike motor based means of transport. Number of customers per day may significantly fluctuate. Capacity to predict count of hourly bike riders can enable companies to control them more effectively and economically.

Bikesharing service provides convenience to lives of citizens and serves as effective system of public transportation. Each station has its dedicated spot to store a bike the same way it may be empty or full at different times.

Development of cycling infrastructure, namely, introduction of urban bike rental, is a popular measure of partial unloading of street and road network from passenger cars, which become common in last decade. Need to effectively manage bikesharing system to cut its operating costs generates many optimisation tasks, including prediction of demand for bikes at each station. Though, real demand on station is difficult to measure due to the limit of its occupancy. When there are no bikes or parking lots available, demand for any of those is not detected.

Prediction of bikesharing trends is crucial not only from business but also planning point of view. Companies engaging in selling bikes must keep track of rental tendencies for determining new features and rearrangement of bikes to place them to the most in-demand areas. Waiting whilst all bikes are rented in specific spot prior to position of extra bikes to that location will lead to profit loss for service provider. This thesis deals with prediction of number of bikes that can be rented per hour and investigates the technique with the minimum error value. The importance of this subject is connected to the rising demand for bikesharing.

2 Objectives and Methodology

2.1 Objectives

The goal of this thesis is using regression and neural network models estimate demand of bikesharing. It will include: Conduct regression analysis, Training of neural network, Validation of findings from both approaches, Determining the accuracy and robustness of results.

2.2 Methodology

In the literature review, there will be a study of professional and scientific sources. Two techniques of supervised learning will be characterised. In the practical part, there will be data preparation and analysis model development and evaluation.

3 Literature Review

3.1 Bikesharing

Over the last few years, public bike rental systems have been the subject of increased attention in the light of emergence of initiatives to increase the use of bicycles, improve quality of connectivity on the “first/last mile” when using other means of transport, reduce harmful effects of vehicles on the atmosphere. The concept of public bikesharing system dates back to “revolutionary” 60s, but evolved very slowly, until the introduction of more advanced ways to track bicycle movements. Their appearance marked beginning of rapid spread of public bike rental systems across Europe and many other countries in that decade. (DEMAIO, 2009)

3.1.1 History of development

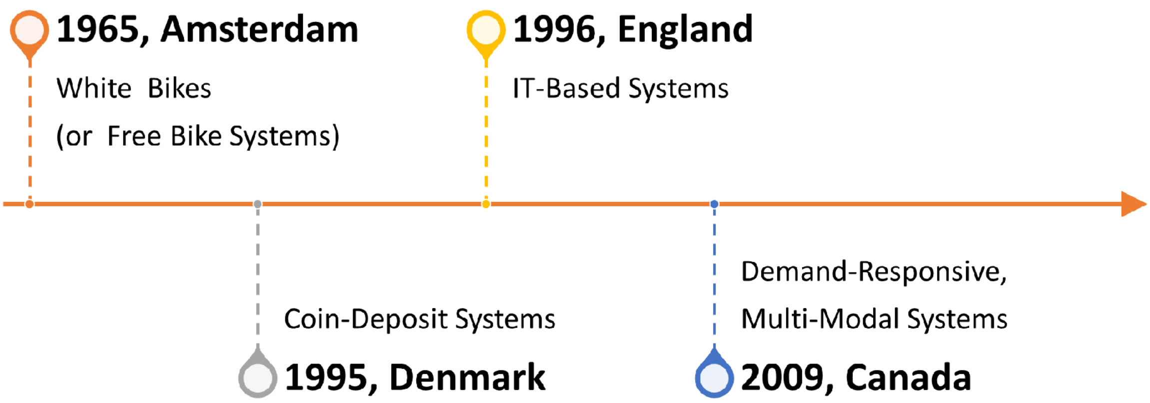

Figure 1: Bikesharing generations. Source: Own processing according to (DEMAIO, 2009) and (SHAHEEN, 2014)

In the development of urban bikesharing systems can be distinguished three generations and discussed potential fourth generation. From the DeMaio, P.’s conversation with Schimmelpennick, L., **first generation** originated in July 28, 1965 in Amsterdam with emergence of “Witte Fietsen” (White Bikes). These were the most common white-coloured bikes offered to public. Anyone who found a bicycle could have ridden it where they wanted to and just left it for next person who wanted it. Everything went wrong: bikes were dumped in canals or kept for themselves. The program was discontinued in a matter of days. (DEMAIO, 2009)

The **second generation** arises from two bike systems in Danish cities of Farsø and Grenå (1991), and Nakskov (1993). ((NIELSEN, 1993), as cited in (DEMAIO, 2009)) However, they were small-scaled programs – in particular, Nakskov had a total of 26 bicycles and 4 bike stations. Only in 1995, in Copenhagen, a full-scale second-generation bikesharing system "Bycyklen" or "City Bikes" was launched, containing many enhancements compared to previous generations. Copenhagen bicycles were specifically engineered for intensive use in urban environments and featured whole rubber tubeless tyres and promotional panels. Bikes could only be taken and left at special stations, leaving coin in the lock as collateral. This system was more formalized than its predecessors, had its own stations, was run by a non-profit organization, and yet the bicycles were subject to theft due to user anonymity. The immediate consequence was the development of a new generation of bike rental systems with better user tracking tools. (DEMAIO, 2009)

The first of its kind **third generation** system was "Bikeabout" at the University of Portsmouth in England (1996), which could be used by students renting a bike with a magnetic stripe card. ((BLACK, 1999), as cited in (DEMAIO, 2009)) This and subsequent systems were equipped with various technological innovations, including racks with electronic locks, telesystems, smart cards and keychains, access via mobile phones, on-board computers. Growth of public bike rentals slowed down in the subsequent years; one to two new programs were introduced per year, these include "Vélo à la Carte" (Rennes, France, 1998) and "Call a Bike" (Munich, Germany, 2000). This continued up to 2005, when the company "JCDecaux" set up "Velo'v" rental system in Lyon, France. ((OBIS, 2009), as cited in (DEMAIO, 2009)) Back then, it was the largest bikesharing system and had significant impact. System numbering 15,000 users and each bicycle being rented on average up to 6.5 times daily, was sighted in Paris. ((HENLEY, 2005), as cited in (DEMAIO, 2009)) Such widespread activities, as well as their impact, surpassed all expectations, changed the course of history of public bikesharing systems and invoked an incredible interest in this mode of transportation all over the world. Apart from Europe, new urban bike rental systems emerged since 2008 in Brazil, Chile, China, New Zealand, South Korea, Taiwan and the United States. Each of these systems became the first third generation system in their country. (DEMAIO, 2009)

Analysts believe that Canadian "BIXI" system, introduced in May 2009, represented the start of **fourth generation** of bikesharing. One of enhancements is portable stations for bicycles, which can be relocated to different spots within the city in response to users' needs. There is potential for future stations to rely on solar panels. On top of that may be considered elimination of rental stations, and bicycles will be possible to pick up and leave on the street using mobile application, as is done in Germany. An important characteristic of fourth generation bike rental systems is integration with other means of transport, including shared payment system and real-time data on all means of transportation, which will result in decrease in the use of personal vehicles in the city. (SHAHEEN, 2014)

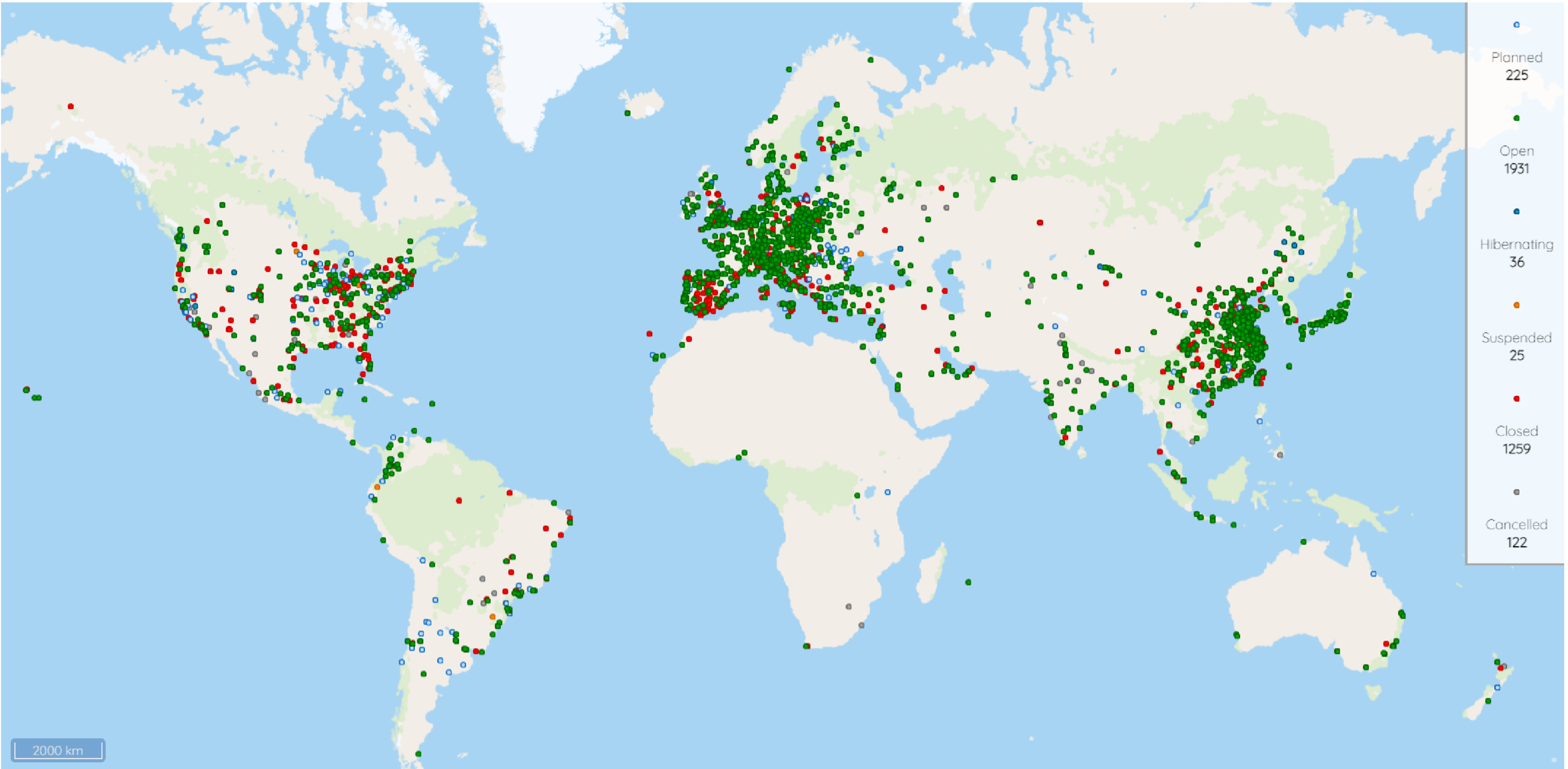

Figure 2: World map of bikesharing. Source: (DEMAIO *et al.*, 2023)

In 2020, there were roughly 1,963 (PBSC Urban Solutions, 2021) such systems in the world. By August 2021, it had grown to 1,999 (PBSC Urban Solutions, 2021). So far (August 2022), there are approximately 1,914 (PBSC Urban Solutions, 2022) systems.

3.1.2 Overview of impact

Operation of public bike rentals has a significant impact on increase in use of bicycles as a means of transport, reduction of emissions into the atmosphere, and improvement of public health's quality. It has resulted in growth in the share of bicycles in public transport by 1 – 1.5 percent in those cities where bicycles were not used extensively in the past. In Barcelona, after the introduction of "Bicing" from 2005 to 2007, the percentage of bike mode increased from 0.75 to 1.76. ((ROMERO, 2008), as cited in (DEMAIO, 2009)) In Paris, the percentage rose from 1 to 2.5 between 2001 and 2007 (introduction of "Vélib"). ((NADAL, 2007), as cited in (DEMAIO, 2009)) Over the years, both cities have improved their bicycle infrastructure, but it is hard to assess how these improvements stimulate use of bike. (DEMAIO, 2009)

Share of bicycles as vehicles in cities with public bikesharing systems is rising due to an increase in the number of rides, better connectivity with other means of transport due to the fact that the first/last mile issue is being addressed, reduced use of personal transport. Meanwhile, as public bikes are becoming the primary mean of transportation for particular categories of rides (up to 50 percent of all rides carried out, as in the case of the "Velo'v" system in Lyon), loss of passenger traffic for public transport is relatively small, since many users of bike rentals, however, do not give up their travel passes. ((BÜHRMANN, 2007), as cited in (DEMAIO, 2009)) According to official data, the number of "Vélib" bike rides in Paris was 50 million within the first two years of system operation. In 2008, 28 percent of those surveyed were inclined to discontinue the use of personal vehicles, in 2009 – already 46 percent. As of 2008, 21 percent of those surveyed reached the underground, train, bus by "Vélib" and 25 percent took "Vélib" on their way back. In 2009, 28 percent resorted to "Vélib" at the beginning as well as end of the ride with transfers to several means of transport. (DEMAIO, 2009)

Many public bikesharing programs are proud of contribution they make to environmental protection and sustainability. According to Montreal "Bixi" management, operation of system prevented emissions of up to three million pounds of greenhouse gases since it was set up in 2009. System launched in Lyon in 2005 protected the atmosphere from 18,600,000 pounds of carbon dioxide. While the impact on public health is yet to be assessed, but the positive impact of bike use on state of health is widely known.

(ANDERSEN *et al.* (2000), CAVILL (2007), SHEPHARD (2008), as cited in DEMAIO (2009))

3.2 Supervised machine learning

Supervisor is defined as someone who is with higher position of authority. If there are doubts, supervisor instructs how to proceed. The same applies to supervised learning – it learns by examples which this supervisor gives (as a teacher in school). Supervised machine learning system requires labelled data from which it develops some knowledge known as model. For instance, if such system is provided a set of photos of people with records about their ethnicity, then model can be trained to identify ethnic origin of person who has never met before in a random photo. In simple terms, model is a function which labels data specific category. This is done by previously collected examples, referred to as training set, using these data as reference information. (SHUKLA, 2018)

The main objective of supervised machine learning algorithms is to maximally reach the level of human expert or commonly accepted truth in outcome prediction. Typical example is to analyse image and determine if the object is cat or dog. In that scenario, expert will be any person who can tell whether the photo depicts a cat or dog. Algorithms use dataset of result (Y) and prospective specs or traits (X) compiled by experts. Algorithm automation is referred to as model. It uses universal statistical approach and provides versatile instance to predict a specific task to be accomplished. (IANSITI, 2020)

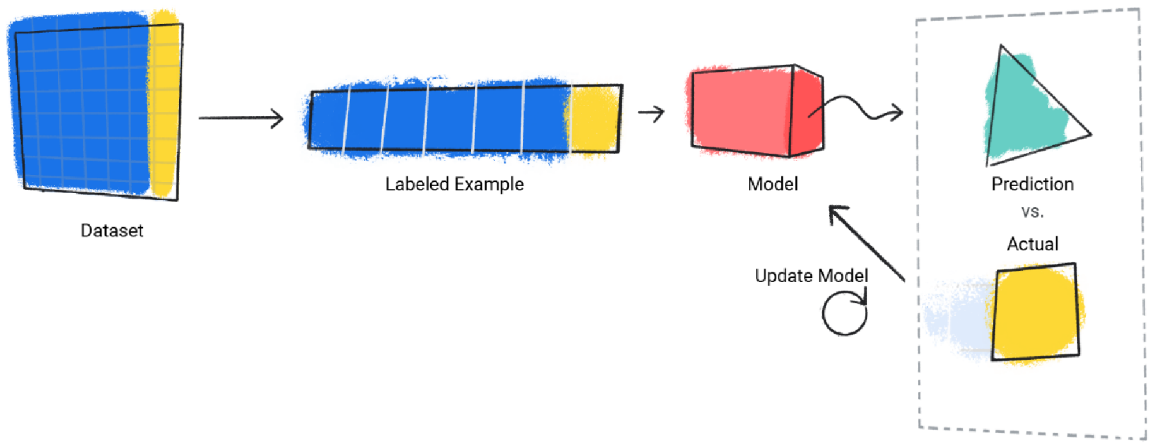

Figure 3: Model training. Source: (Google Developers, 2022)

Since predictive models are clearly instructed on what and how to learn, training process of predictive model is referred to as supervised learning. By supervision here means not participation of human being, but instead the event where target values enable learner to determine how well he or she has grasped the task. From a more formal point of view, supervised learning algorithm aims at optimisation of function (model) to find compound of trait values that lead to target result for a particular dataset. It tries to detect and model relation of target (predicted) to other traits. (LANTZ, 2019)

In most cases, validation of supervised learning algorithms is easier because labels can be used to evaluate quality of predictions obtained from model. Access to target parameters also facilitates process of model training. Although creation of dataset with labels often requires large investment of time at preliminary preparation stage, it greatly simplifies construction and validation of models. At the same time, by figuring out what input data the model will take and what data it will return, number of possible approaches to modeling will be significantly reduced. (AMEISEN, 2020)

3.2.1 Machine learning process

Whatever the task, any ML algorithm can be broken down into the following steps:

Data analysis

Presenting a dataset as a black box is not the best practice. Before proceeding to model training, it is necessary to analyse and visualise the data to understand what features give it the predictive ability, which will help to reasonably design the features and identify potential problems. If there are numerical features in data, feature distribution histogram is plotted to get an idea of the range and frequency of different values. (CHOLLET, 2021)

Several features may be strongly correlated and hence be unnecessary to some extent. In such situations, it is recommended to use dimension reduction methods to compress them into subspace of smaller dimensions. Its advantage is that less memory is needed, as a result, training algorithm can be executed a lot faster. (RASCHKA, 2019)

Correlation between two variables is a number that shows how close their relationship is to a linear relationship. It is usually termed as Pearson correlation coefficient, which was introduced in the 20th century by mathematician Karl Pearson. Correlation is in the range of -1 and +1. It is described as follows:

$$\rho_{x,y} = Corr(x,y) = \frac{Cov(x,y)}{\sigma_x \sigma_y} \tag{1}$$

where $\rho_{x,y}$ is Pearson correlation coefficient, σ_x and σ_y are standard deviations of x and y respectively, and $Cov(x, y)$ is covariance function. There are a number of empirical rules for interpretating strength of correlation. "Weak" status is assigned to values from 0.1 to 0.3, "moderate" to values from 0.3 to 0.5, and "strong" for values above 0.5 (this is also true for similar negative correlation ranges). (LANTZ, 2019)

Technique selection

The decision on technique is determined by the problem to be solved. There are many of them, few of which will be explored later in this paper. When prediction is inquired, supervised learning techniques can be deployed. Apart from their primary task, techniques are distinct in other factors, namely their capacity to analyse various types of data, and the format of the output results. (NG, 2017)

In classification task, decision trees provide to construct comprehensible models, whereas models of neural networks are known to be difficult to interpret. This can be crucial point in development of a creditworthiness model, as the legislation often expects the applicant to be informed of the reasons for the credit rejection. Although neural network better predicts the likelihood of credit default, but its predictions are impossible to explain, in this case it is not efficient. (LANTZ, 2019)

Train-Test Split

To confirm that machine learning technique operates well both on the training set and new data, dataset should also be split into two subsets: train and test sets. Training set is used to train and improve machine learning model, while test set is stored to make prediction. In practice, 60:40, 70:30 or 80:20 split proportions are usually used, determined by the size of the original dataset. (RASCHKA, 2019) However, splitting 30% of the data as a test set is the gold standard for evaluating model performance. So there will be an independent dataset on which the model is not training. (AKALIN, 2020)

Cross-validation can help in cases where there is too little data and it may be too expensive to store much of the dataset as a test set. It operates by splitting data into arbitrary selection of k subsets called k-folds. As seen in a five-fold cross-validation-check of 100 data points, five folds having 20 data points each, will be created. (AKALIN, 2020)

Model training, prediction

Once the data is ready for training, there is likely to be an insight of what can be derived from it. Selected technique, determined by specific task of machine learning, provides data in the form of model. (LANTZ, 2019)

Model predicts based on test data, then these predictions can be compared with real ones and thus model quality is assessed. By comparing result prediction of algorithm model, it can be decided if difference in the number of errors between model and actual results is acceptable. If not, another approach may be chosen, obtained additional data or identified other features that could be valuable for a more precise prediction. (IANSITI, 2020)

Upon completion of these steps, the model can be applied to given task. According to circumstances, it may be used to provide estimated data of forecasts (potentially in real time), forecast financial data, analytical review for marketing, automate some tasks. Successes and failures of the model can also provide additional data for further training. (LANTZ, 2019)

3.2.2 Applications

Machine learning achieves the best results where it supplements but not substitutes the specialist knowledge of an expert in a specific area. Doctors use ML in the fight against cancer; it helps engineers and programmers in building more intelligent houses and cars, and social scientists in accumulating knowledge on how society operates. For this purpose, machine learning is used in many enterprises, research laboratories, clinics and government organisations. Wherever data is generated or accumulated, at least one ML algorithm is likely to be used. (LANTZ, 2019)

Following examples of supervised machine learning can be found:

- Use image pixels to determine whether the cat is present or absent.
- Use a list of favourite movies to recommend movies that one might like.
- Use words in a message to predict if the writer is happy or upset.
- Use meteorological data for rainfall probability prediction.
- Use of vehicle engine sensors to determine ideal settings.
- Use news to forecast future stock prices.

- Use of audio file to extract transcription of speech contained in it.

Either way, the machine learning technique seeks to identify such patterns between the two data sets so that one can predict the other. (TRASK, 2019)

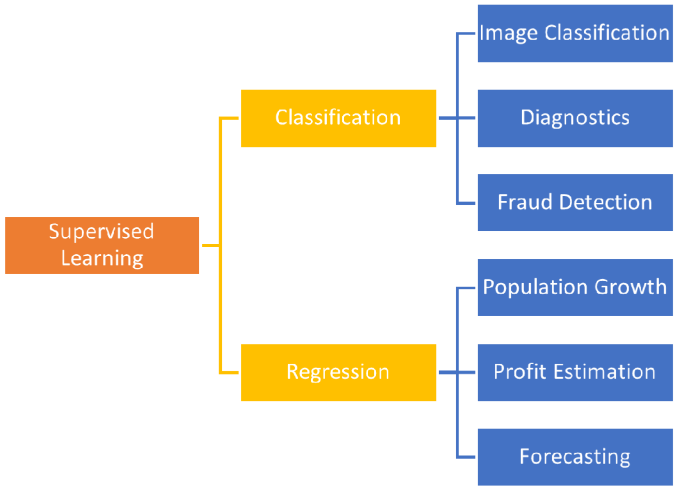

Figure 4: Applications of supervised learning. Source: Own processing according to (MORDENSKY *et al.*, 2022)

Whenever an email is marked as spam, machine learning algorithms of email service update their models to detect the newest instances of fraud. The ability of Facebook or Baidu to recommend friends names that may show up on recently uploaded photos, guided by previous photo labels. Companies issuing credit cards or delivering payment platforms, make a choice about whether or not to approve a transaction based on past shopping patterns that auto-generate labelled data. Netflix applies supervised learning for various scenarios. In terms of recommendations, it employs labelled datasets containing actions and results (e.g., selected and liked movies) from people who according to algorithm are similar to this user. Large dataset on user selections, adjusted in line with his/her characteristics and decision-making background, can provide worthwhile recommendations. This kind of co-filtering algorithm is implemented in wide range of recommendations, along with the engines for Amazon purchases and Airbnb matching. (IANSITI, 2020)

3.3 Types of problems

Two main tasks in case of supervised learning are classification problem with categorical target variable and regression problem, where variable is numerical. (BRINK *et al.*, 2016)

3.3.1 Classification problem

Classification is a type of supervised machine learning, the purpose of which is to identify common patterns in instances consisting of independent variables and their relationship to one or another category. The error between estimated and real categories of training data can be reduced to minimum by training of classifier which can then be used to categorise new instances according to patterns discovered in training phase. (BENGFORT *et al.*, 2018)

Classification implies prediction of categories of samples in accordance with their features. (Is there a way to determine what the object is – passenger car or truck – from number of wheels, weight and maximum speed?) (HURBANS, 2020)

Discussion of most machine learning models should be initiated with classification. It is a type of supervised learning that uses data to select names, values and categories. For instance, neural network can scan photos looking for an image of shoe. There are two types of classification:

- **binomial** – if one of two categories is selected (coffee or tea);
- **multiple** – if more than two options exist. (LEA, 2018)

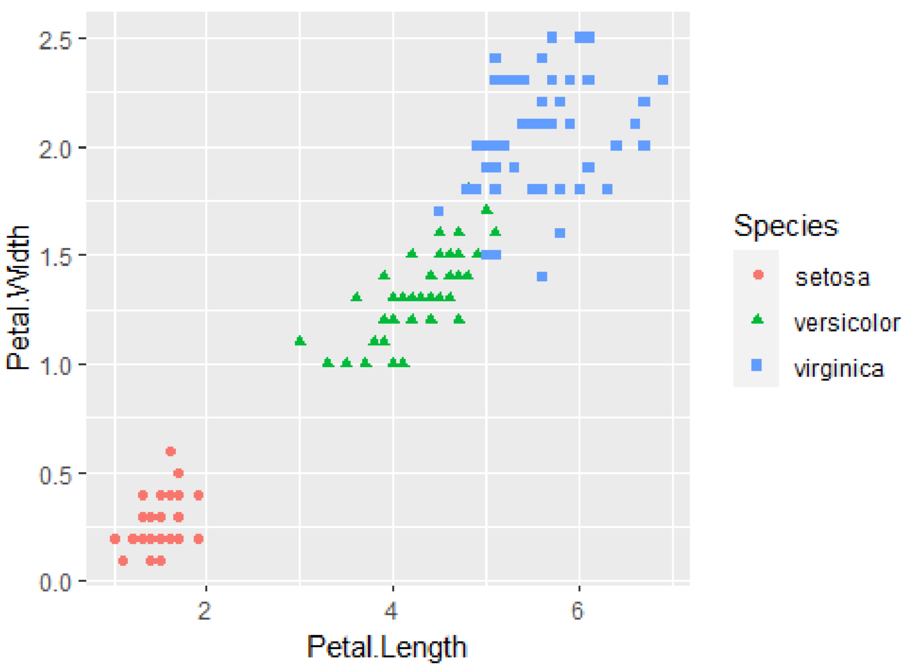

Figure 5: Example of classification task. Source: Own processing using Edgar Anderson's Iris Data (ANDERSON, 1935)

Figure 5 illustrates multiple classification of three species: setosa, versicolor and virginica. Variables are split into three definite areas by colour (red, green and blue) and shape (circles, triangles and squares). It is worth noting that such relationship can contain errors.

3.3.2 Regression problem

Regression is a type of supervised machine learning, the purpose of which is to identify common patterns in instances consisting of independent variables and their belonging to one or another target value. The error between estimated and real categories of training data can be reduced to minimum by training of regressor which can then be used to set target values to new instances according to patterns discovered in training phase. (BENGFORT *et al.*, 2018)

Regression implies tracing a line across a set of data points in a way that it complies to maximum extent with general form of data. It may be used, for instance, to establish relationship tendencies between marketing shares and sales. (Does promotion of goods through online advertising have a direct connection to actual sales?) It may also serve to identify influencing factors. (Does time have a direct connection to value of cryptocurrency, and will there be an exponential growth in value over time?) (HURBANS, 2020)

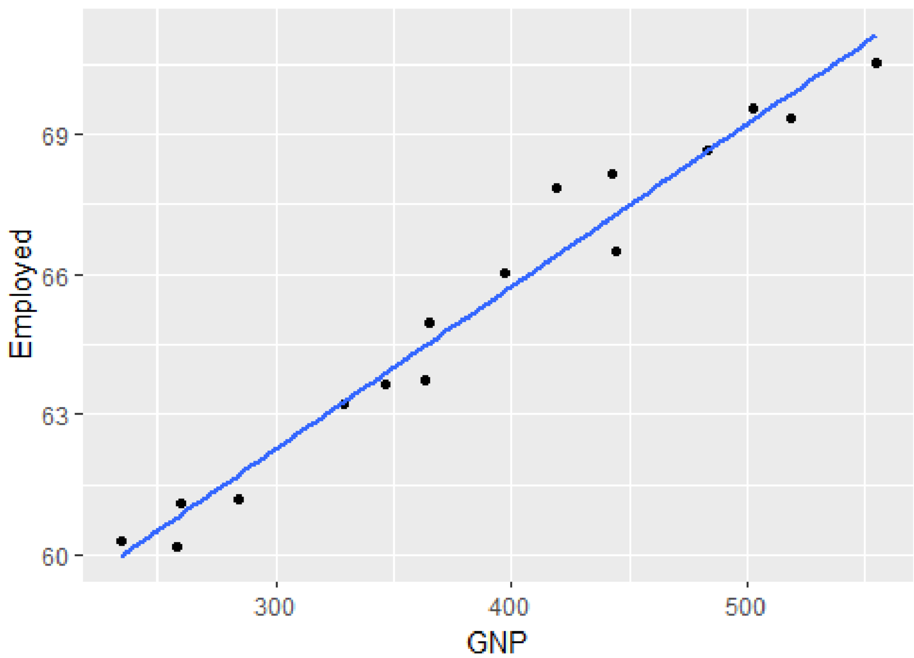

Figure 6: Example of regression task. Source: Own processing using Longley's Economic Regression Data (LONGLEY, 1967)

Classification focuses on prediction of discrete value (circle or square), while regression models give a prediction of continuous values. For instance, regression analysis allows to predict average house price, depending on cost of all houses in the vicinity and its surroundings. (LEA, 2018)

3.4 Machine learning techniques

Machine learning techniques are mainly used in prediction, classification and clustering. (SCHUTT, 2014) Prediction of supervised learning algorithm can result in a category (dog or cat), in which case logistic regression is used, or numeric value (English proficiency assessment), in which case linear regression is used. Other noteworthy approaches, depending on the depth and breadth of available data and type of problem to be solved, include support vector machines, k-Nearest Neighbours (k-NN), random forest and neural networks. (IANSITI, 2020)

3.4.1 Regression analysis

Regression includes determining the relationship between one dependent numeric variable (predicted value) and one or several independent numeric variables (predictors). As the name suggests, dependent variable is defined by the value of the independent variable(s). The most basic forms of regression suppose that the relationship between independent and dependent variables is linear. Data models of regression equations use slope-intercept equations of the form:

$$y = a + bx \tag{2}$$

where y denotes dependent variable and x denotes independent variable, b is slope coefficient and a is intercept coefficient. The work of machine lies in identification of values a and b in the way that line best associates given values of x with y values. Regression analysis is applied to solve different problems – it is one of the most popular techniques of machine learning. It can be deployed not only to predict the future but also to explain the past, and applicable to almost any problem. (LANTZ, 2019)

Methods

Regression analysis implies more than one technique. Instead, it is a general name for many methods which can be fitted to almost any machine learning problem. The simplest models of **linear regression** are those in which straight lines are used. If there are two or more independent variables it is the case of **multiple regression**. The two methods consider just one dependent variable, that is measured continuously.

Alternatively, regression may be applicable to other types of dependent variables and even possible to be employed to few classification tasks. To be more explicit, **logistic regression** is applied to model result of an event in binary form (0 or 1) and **Poisson regression**, under the name of French mathematician Siméon Poisson, models result as whole numbers (i.e. count data). Many specialised regression methods are classified as **Generalized Linear Models (GLM)**. Using GLM, it is possible to extend linear models to other patterns with the help of link function that defines more sophisticated forms of relationship between x and y, which enables applying regression to nearly all types of data. (LANTZ, 2019)

Poisson regression

Poisson regression is applied when the count dependent variable is predicted for set of continuous and/or categorial dependent variables. For data with Poisson distribution, a lower variance can be expected to be associated with a lower mean. Variance heterogeneity is not an issue for Poisson regression, as opposed to standard regression. (KABACOFF, 2022)

Poisson regression assumptions:

- Poisson response – response variable must be count data that includes integers and cannot contain negative values.
- Independence – each observation in the data set should be independent of each other which means that one observation should not give any information about another one.
- Mean = variance – for Poisson distribution, the variance is the same as the mean.
- Linearity – natural logarithm of mean rate should be a linear function from explanatory variable. (ROBACK, 2021)

Model implies that response variable has a Poisson distribution and that it is modelled as a natural logarithm of conditional mean:

$$\log_e(\lambda) = \beta_0 + \sum_{i=1}^{n} \beta_i X_i \tag{3}$$

where λ is the mean (and variance) of dependent variable, β_0 and β_i are regression coefficients. Poisson distribution assumes equality of variance and mean. Excessive variance in Poisson regression is noted when the observed variance of the dependent variable is greater than expected, based on distribution properties. It often occurs when working with count data and may adversely affect interpretation of results. In such cases Poisson regression is replaced by Quasipoisson to construct the model. Parameter estimates for this approach are exactly the same to those obtained in the Poisson regression, but standard errors are much larger. (KABACOFF, 2022)

Training

Poisson regression model is trained on training set by estimating values of regression coefficients to approximate real value of response variable. For that, method called Maximum Likelihood Estimation (MLE) is used. (DATE, 2019) It evaluates unknown parameters by maximizing the probability function, resulting in model parameter values, that make the outcome "closer" to the real. Likelihood determined by the whole dataset equals to the product of individual likelihoods, which is maximum exactly when selected such coefficients, which minimise the sum of squared of errors. In other words, minimising the sum of squared errors is the same as maximising the likelihood of observed data. (GRUS, 2019)

3.4.2 Neural networks

Neural network (NN) models relationships between multiple input signals and output signal. In general, it is universal machine learning technique that can be adopted to solve almost any problem: classification, numerical prediction and even pattern recognition. (LANTZ, 2019)

Basic architecture

A classic neural network is comprised of several tens to hundreds, thousands, or even millions of artificial neurons known as units, organised in multiple layers, each of which is connected to the other layers on both sides. **Input units** are intended to accept different types of information that the network will try to discover, detect, or alternatively process. Other units are located on the opposite side of the network and indicate how they react to detected information. These units are called **output units**. Between the input and output units, there are single or multiple layers of **hidden units** that together make up most of the artificial brain. In majority of neural networks, each unit is connected to each other. The links between the two units are displayed by a number known as a **weight** which may be positive or negative. The larger the number, the greater the effect of one unit over another. (WOODFORD, 2021)

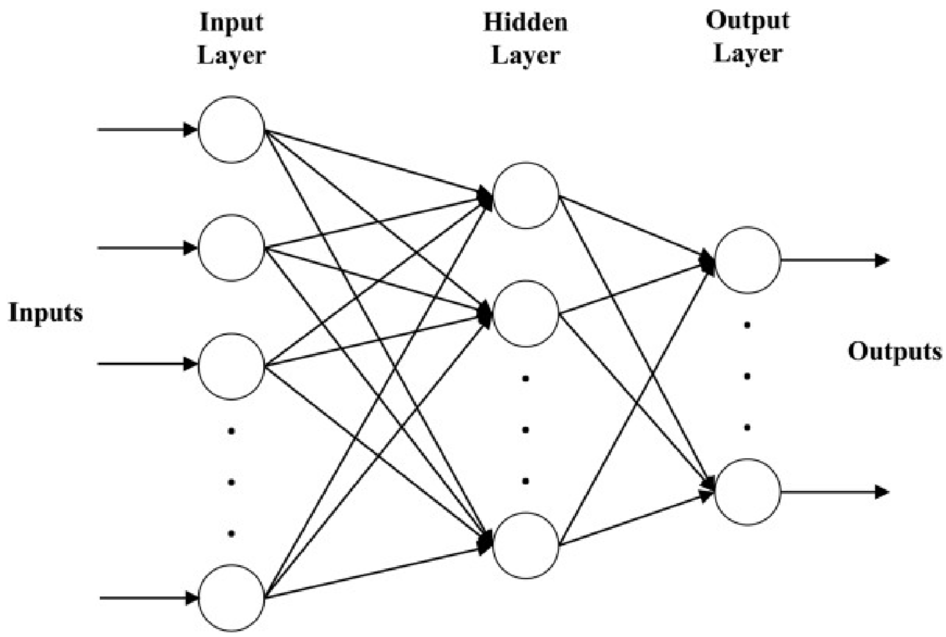

Figure 7: Neural network architecture. Source: (VENKATESWARLU, 2022)

There are various ways to regulate the complexity of the neural network: the number of hidden layers, number of elements in each hidden layer and regularization. (MÜLLER, 2016) As a rule, the larger and more complex the network, the more implicit patterns it can find and has broader decision boundaries. Single-layer networks can be used for basic classification of patterns, especially those that are linearly separable, however, many machine learning tasks require more complex networks. Number of input nodes is based on the amount of features in the input data. In the same way, number of output nodes is based on the amount of modelled outcomes or the amount of resulting class levels. On another note, the decision on the number of hidden nodes is made by user before starting to train the model. (LANTZ, 2019)

Normalisation

As a rule, it is risky to pass on to a neural network data that accepts very high values (e.g., integers with much larger number of significant digits than the initial weights of the network) or heterogeneous data (e.g., these data in which the features are defined by values in different ranges). This can lead to significant changes in the gradient, which will hinder the convergence of the network. To facilitate network learning, data should:

- have small values – usually values should be between 0 and 1;
- be homogeneous – that is, all features should be within approximately the same range. (CHOLLET, 2021)

Features with completely different scales cause confusion of model. Feature with values between 0 and 1000 may get offset weight if other features have values between 0 and 100. To avoid this problem, features can be scaled to have the same range, which is called normalisation. The most straightforward approach to normalisation is known as **MinMax**. Values are converted to range from zero to one, where zero is minimum and one is maximum value, and remaining values are set between them.

$$x_{scaled} = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{4}$$

When MinMax is used, all features are within single range. Furthermore, outliers will make non-outliers to fit closer together, resulting in smaller areas in the 0-1 range. Besides, values can be normalised based on their individual distance to the mean and standard deviation of feature, where mean value is converted to zero and standard deviation to one, respectively. This approach to normalisation is referred to as **z-score**.

$$x_{scaled} = \frac{x - mean}{std.dev.} \tag{5}$$

When z-score is used, outliers make less impact as they do not induce compression, and non-outliers ranges have less difference between features. (FERREIRA FILHO, 2021)

Training

Key characteristic of neural networks is that their weights are initialised at random before starting training that has an impact on the learning process of the model. So even with the same parameters, completely different models can be derived, by specifying different start values to generator of pseudorandom numbers. (MÜLLER, 2016)

There are two ways in which information flows across the neural network. When it learns (trains) or runs in general (after training), the information patterns are delivered to the network through the input units that initiate layers of hidden units, which, consequently enter the output units. This overall concept is referred to as a **feedforward network**. Each unit gets input signals from the left, which are then multiplied by the weights of the connections they have already surpassed. Each unit sums all the input signals and if the total exceeds some threshold, the unit dismisses and calls the connected units (the right-hand ones). Feedback process of neural networks learning known as the "backprop network" (short for **backpropagation method**). This method is based on comparison of the network output with the output it was supposed to generate, and use of the difference between them to change the connections' weights among the units in the network, operating from the output units via the hidden units to the input units – namely, moving back. As this transition progresses, network learns by decreasing the difference between the real and expected output. (WOODFORD, 2021)

3.4.3 Other techniques

Decision Tree

Decision tree machine learning technique is powerful classifier that uses tree structure for modelling relations between features and possible outcomes. It allows to make complex decisions by choosing from many simple options. (LANTZ, 2019) Decision trees can be applied to solve problems of classification and regression. (BASHA, 2019) They are basically asking questions and constructing a hierarchy of rules "if ... then" to take a decision. Each node of the tree is either question, or closing node (it is also known as leaf) with the answer. Edges connect top nodes with bottom ones. (MÜLLER, 2016)

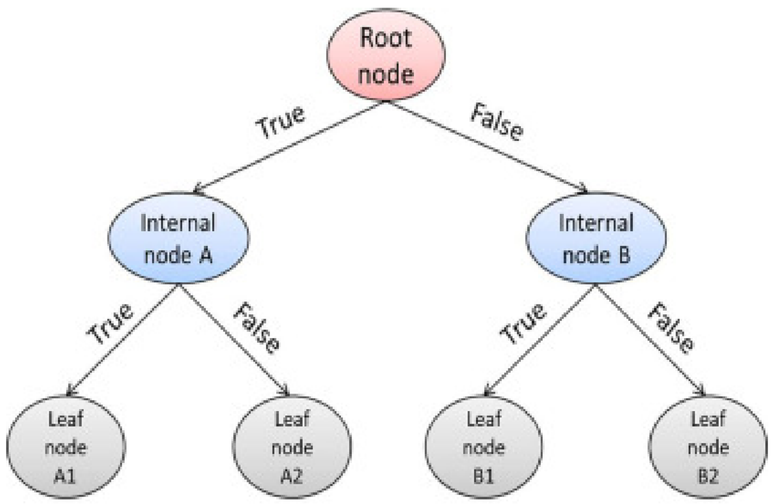

Figure 8: Example of decision tree. Source: (RANI *et al.*, 2022)

Decision tree model is easy to visualise, which can be an advantage for non-specialists to understand. They have good performance even when features are measured in totally different scales, or when data is a mixture of binary and continuous features. The main disadvantage of decision trees is their poor capacity to generalise. (MÜLLER, 2016) Tasks in which data have many nominal features with many levels or numerical features, can lead to a huge number of decisions and excessively complex trees. (LANTZ, 2019)

Random Forests

Random forests technique was proposed by Leo Breiman and Adele Cutler. It combines the basic principles of bagging with random selection of features, allowing for greater diversity in decision tree models. After the generation of the tree ensemble (forest), model combines forecasts of individual trees by voting. (LANTZ, 2019)

If results of multiple trees are obtained in different ways, but consistent among themselves, and just one tree does not fit into this pattern, it will naturally make the decision of the majority. This model has smaller variance than the one decision tree, which can be strongly biased. (LEA, 2018)

k-Nearest Neighbors

The k-Nearest Neighbor (k-NN) is one of the simplest and apparent machine learning techniques. It solves regression and classification types of machine learning problems. (SUBASI, 2022) Model is built by storing the training dataset. For predicting new data point, technique looks for the closest points of the training set, i.e. finds "nearest neighbours". Basically, two main parameters are specified – number of neighbours and measure of the distance between data points. In fact, having a few neighbors (e.g., 3-5) demonstrates good performance. The default distance measure is Euclidean distance, which works well in many situations. Although the KNN is not difficult to interpret, in reality it is not frequently applied because of its computational speed and its incompetence in dealing with large number of features. (MÜLLER, 2016)

Support Vector Machine

Support Vector Machines (SVMs) are derived from theory of statistical learning and were established by Vladimir Vapnik during the 1990s. This supervised learning technique solves classification and regression types of problems. (SUBASI, 2022)

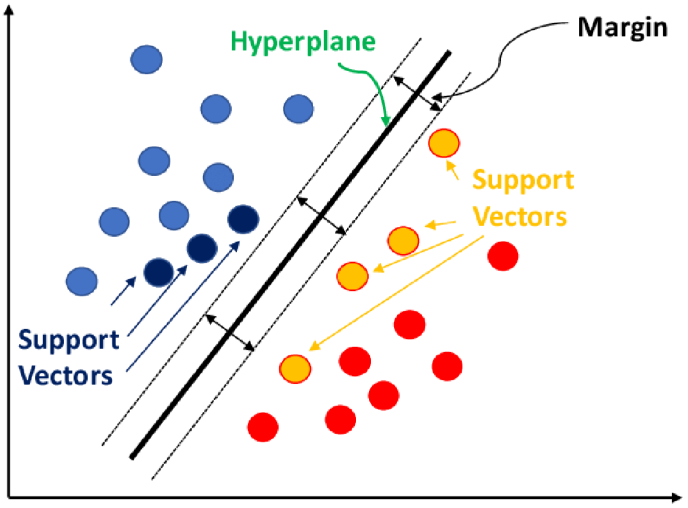

Figure 9: Illustration of support vector machine. Source: (RANI *et al.*, 2022)

SVM examines the outlying borders and draws edges, often called hyperplanes, that separate the two classes. Poor quality decision borders may cause new data point to be incorrectly classified. Outlying data points are of assistance in defining boundaries known as support vectors, and they are more likely to ignore learning data. Thus, hyperplane is made of by adding the shortest positive distances to the shortest negative points. (BASHA, 2019)

3.5 Performance assessment

Measures allow to determine how well approximate values correspond to modelled results. (EL HAMI, 2020) Because regressors predict values, the regular approach to estimate proximity of prediction from real value is to calculate the error between them:

$$error = y - \hat{y} \tag{6}$$

where $\hat{y}$ is predicted value and y is real value. One error value is provided by evaluation for each row. As a general rule, all these errors are summarised by some average value. Below are couple of frequent methods to measure this. (FERREIRA FILHO, 2021)

3.5.1 Mean Absolute Error

The most widely used average is arithmetic mean. Though, in the calculation of arithmetic mean, positive and negative errors offset one another conveying a misleading perception of accuracy. This can be prevented with MAE, arithmetic mean of all absolute error values, in other words, arithmetic mean of all error values modulo. (FERREIRA FILHO, 2021)

$$MAE = \frac{\sum_{i=1}^{n} |error_i|}{n} \tag{7}$$

This measure is comparable with RMSE (see further). However, it is more reliable due to its low sensitivity to extreme values. (EL HAMI, 2020)

3.5.2 Mean Squared Error

Mean Squared Error represents the mean square value of forecasting errors:

$$MSE = \frac{\sum_{i=1}^{n} (error_i)^2}{n} \tag{8}$$

It measures the mean square error of the discrepancy between the predicted values and the test data. Small MSE value indicates that the predicted values are approximate to real ones. (EL HAMI, 2020)

This measure is used to assess the quality of the model in training. In essence, the best outcome is provided by the lowest value. It is worth noting that MSE strongly depends on the initial weights. (SCHNEIDER, 2022)

3.5.3 Root Mean Square Error

Sometimes, one highly inaccurate prediction may cause disaster. Taking into account sensitivity to extreme errors, arithmetic mean of squared errors can be computed so that evaluation rejects them more explicitly. (FERREIRA FILHO, 2021)

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n} (error_i)^2}{n}} \tag{9}$$

In this case, negative numbers are not an issue, because negative number squared is still positive number. Moreover, RMSE returns the value to the original error scale. MAE and RMSE may provide a glimpse of capability of model to predict the future. The more model is aggregated with new data, the more these estimates will approximate errors in future predictions. (FERREIRA FILHO, 2021)

This measure gives insight on the short-run quality of the model, making it possible to compare the difference between the predicted and real value for each individual case. The lower the value, the higher is performance of the model. ((MA, 1984), as cited in (KAMBEZIDIS, 2012))

4 Practical Part

4.1 Dataset description

Dataset is based on real data from American company and available at UC Irvine Machine Learning Repository. (FANAEE-T, 2013) It represents number of bikes rented at particular time of day in 2011-2012 and contains 17,379 rows and 17 columns. Selection of unprocessed features includes weather conditions (temperature, humidity and wind speed) and day type (holiday/work day). Following is the complete list of features for each day of rental (as they were specified in data source):

instant: record index
dteday: date
season: (1 – winter, 2 – spring, 3 – summer, 4 – autumn)
yr: year (0 – 2011, 1 – 2012)
mnth: month (1 to 12 – January to December)
hr: hour (0 to 23)
holiday: (0 – not holiday, 1 – holiday)
weekday: day of the week (1 – Monday, 2 – Tuesday, 3 – Wednesday, 4 – Thursday, 5 – Friday, 6 – Saturday, 0 – Sunday)
workingday: (0 – weekend or holiday, 1 – neither weekend nor holiday)
weathersit: weather situation

1. Clear, Few clouds, Partly cloudy, Partly cloudy
2. Mist + Cloudy, Mist + Broken clouds, Mist + Few clouds, Mist
3. Light Snow, Light Rain + Thunderstorm + Scattered clouds, Light Rain + Scattered clouds
4. Heavy Rain + Ice Pallets + Thunderstorm + Mist, Snow + Fog

temp: normalised temperature in Celsius (values are derived via MinMax, where minimum temperature is -8, maximum temperature is +39)
atemp: normalised feeling temperature in Celsius (values are derived via MinMax, where minimum temperature is -16, maximum temperature is +50)
hum: normalised humidity (values are divided to max of 100)
windspeed (mph): normalised wind speed (values are divided to max of 67)
casual: count of casual users
registered: count of registered users
cnt: count of total rental bikes including both casual and registered

4.1.1 Data preprocessing

Before applying supervised machine learning techniques, it is necessary to make some preparations on raw data. Dataset is loaded to bikesharing_hourly data frame from hour.csv file using `read.csv` function:

```
bikesharing_hourly <- read.csv("hour.csv", header = TRUE)
head(bikesharing_hourly)
str(bikesharing_hourly)
```

`head` function prints first six rows to make sure that data was loaded correctly.

	instant	dteday	season	yr	mnth	hr	holiday	weekday	workingday	weathersit	temp	atemp	hum	windspeed	casual	registered	cnt
1	1	2011-01-01	1	0	1	0	0	6	0	1	0.24	0.2879	0.81	0.0000	3	13	16
2	2	2011-01-01	1	0	1	1	0	6	0	1	0.22	0.2727	0.80	0.0000	8	32	40
3	3	2011-01-01	1	0	1	2	0	6	0	1	0.22	0.2727	0.80	0.0000	5	27	32
4	4	2011-01-01	1	0	1	3	0	6	0	1	0.24	0.2879	0.75	0.0000	3	10	13
5	5	2011-01-01	1	0	1	4	0	6	0	1	0.24	0.2879	0.75	0.0000	0	1	1
6	6	2011-01-01	1	0	1	5	0	6	0	2	0.24	0.2576	0.75	0.0896	0	1	1

Figure 10: Output from head function

`str` function displays each variable in a row, followed by its data type and first few values. Dates are defined as character strings, the rest set of variables are either integer (discrete) or numerical (continuous). Season, year, month, hour, holiday, weekday, working day and weather situation are listed as integers but there are only number of values that they can take, so preferred way is to represent them as categorical variables.

```
'data.frame':	17379 obs. of  17 variables:
 $ instant   : int  1 2 3 4 5 6 7 8 9 10 ...
 $ dteday    : chr  "2011-01-01" "2011-01-01" "2011-01-01" "2011-01-01" ...
 $ season    : int  1 1 1 1 1 1 1 1 1 1 ...
 $ yr        : int  0 0 0 0 0 0 0 0 0 0 ...
 $ mnth      : int  1 1 1 1 1 1 1 1 1 1 ...
 $ hr        : int  0 1 2 3 4 5 6 7 8 9 ...
 $ holiday   : int  0 0 0 0 0 0 0 0 0 0 ...
 $ weekday   : int  6 6 6 6 6 6 6 6 6 6 ...
 $ workingday: int  0 0 0 0 0 0 0 0 0 0 ...
 $ weathersit: int  1 1 1 1 1 2 1 1 1 1 ...
 $ temp      : num  0.24 0.22 0.22 0.24 0.24 0.24 0.22 0.2 0.24 0.32 ...
 $ atemp     : num  0.288 0.273 0.273 0.288 0.288 ...
 $ hum       : num  0.81 0.8 0.8 0.75 0.75 0.75 0.8 0.86 0.75 0.76 ...
 $ windspeed : num  0 0 0 0 0 0.0896 0 0 0 0 ...
 $ casual    : int  3 8 5 3 0 0 2 1 1 8 ...
 $ registered: int  13 32 27 10 1 1 0 2 7 6 ...
 $ cnt       : int  16 40 32 13 1 1 2 3 8 14 ...
```

Figure 11: Structure of raw data

For further analysis, dates are converted to calendar date format, categorical variables are converted to factors and Boolean values will be converted to logical types. Weather situation data is converted to character string for visualisation according to numeric code.

```r
bikesharing_hourly <- bikesharing_hourly %>%
  mutate(dteday = as.Date(dteday),
         season = factor(season, levels = c(1, 2, 3, 4),
                         labels = c("winter","spring", "summer", "autumn")),
         yr = factor(yr, levels = c(0, 1), labels = c(2011, 2012)),
         mnth = factor(mnth, levels = c(1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12), labels = c("January", "February", "March", "April", "May", "June", "July", "August", "September", "October", "November", "December")),
         hr = factor(hr),
         holiday = as.logical(holiday),
         weekday = factor(weekday, levels = c(1, 2, 3, 4, 5, 6, 0), labels = c("Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday", "Sunday")),
         workingday = as.logical(workingday),
         weathersit = ifelse(weathersit == 1, "Clear to partly cloudy", ifelse(weathersit == 2, "Misty", ifelse(weathersit == 3, "Light Precipitation", "Heavy Precipitation")))
         )
```

```
'data.frame':	17379 obs. of  17 variables:
 $ instant   : int  1 2 3 4 5 6 7 8 9 10 ...
 $ dteday    : Date, format: "2011-01-01" "2011-01-01" "2011-01-01" "2011-01-01" ...
 $ season    : Factor w/ 4 levels "winter","spring",..: 1 1 1 1 1 1 1 1 1 1 ...
 $ yr        : Factor w/ 2 levels "2011","2012": 1 1 1 1 1 1 1 1 1 1 ...
 $ mnth      : Factor w/ 12 levels "January","February",..: 1 1 1 1 1 1 1 1 1 1 ...
 $ hr        : Factor w/ 24 levels "0","1","2","3",..: 1 2 3 4 5 6 7 8 9 10 ...
 $ holiday   : logi  FALSE FALSE FALSE FALSE FALSE FALSE ...
 $ weekday   : Factor w/ 7 levels "Monday","Tuesday",..: 6 6 6 6 6 6 6 6 6 6 ...
 $ workingday: logi  FALSE FALSE FALSE FALSE FALSE FALSE ...
 $ weathersit: chr  "Clear to partly cloudy" "Clear to partly cloudy" "Clear to partly cloudy" "Clear to partly cloudy" ...
 $ temp      : num  0.24 0.22 0.22 0.24 0.24 0.24 0.22 0.2 0.24 0.32 ...
 $ atemp     : num  0.288 0.273 0.273 0.288 0.288 ...
 $ hum       : num  0.81 0.8 0.8 0.75 0.75 0.75 0.8 0.86 0.75 0.76 ...
 $ windspeed : num  0 0 0 0 0 0.0896 0 0 0 0 ...
 $ casual    : int  3 8 5 3 0 0 2 1 1 8 ...
 $ registered: int  13 32 27 10 1 1 0 2 7 6 ...
 $ cnt       : int  16 40 32 13 1 1 2 3 8 14 ...
```

Figure 12: Structure of adjusted data

4.2 Data Analysis

4.2.1 Numerical summary

Measures of centre (mean, median) along with five-number summary are output of below function.

```
summary(bikesharing_hourly)
```

```
      temp           atemp             hum           windspeed          casual         registered         cnt       
 Min.   :0.020   Min.   :0.0000   Min.   :0.0000   Min.   :0.0000   Min.   :  0.00   Min.   :  0.0   Min.   :  1.0  
 1st Qu.:0.340   1st Qu.:0.3333   1st Qu.:0.4800   1st Qu.:0.1045   1st Qu.:  4.00   1st Qu.: 34.0   1st Qu.: 40.0  
 Median :0.500   Median :0.4848   Median :0.6300   Median :0.1940   Median : 17.00   Median :115.0   Median :142.0  
 Mean   :0.497   Mean   :0.4758   Mean   :0.6272   Mean   :0.1901   Mean   : 35.68   Mean   :153.8   Mean   :189.5  
 3rd Qu.:0.660   3rd Qu.:0.6212   3rd Qu.:0.7800   3rd Qu.:0.2537   3rd Qu.: 48.00   3rd Qu.:220.0   3rd Qu.:281.0  
 Max.   :1.000   Max.   :1.0000   Max.   :1.0000   Max.   :0.8507   Max.   :367.00   Max.   :886.0   Max.   :977.0  
```

Figure 13: Summary statistics

Variance and standard deviation are measures of spread of distribution printed using functions `var` and `sd`, respectively.

```
apply(bikesharing_hourly[, c(11:14, 17)], 2, var)
```

```
        temp        atemp          hum    windspeed          cnt 
3.707786e-02 2.953250e-02 3.722192e-02 1.496713e-02 3.290146e+04 
```

Figure 14: Variances

Count of total rental bikes varies significantly from its mean. Temperature, feeling temperature, humidity and windspeed values are close to their mean thanks to provided normalisation.

```
apply(bikesharing_hourly[, c(11:14, 17)], 2, sd)
```

```
       temp       atemp         hum   windspeed         cnt 
  0.1925561   0.1718502   0.1929298   0.1223402 181.3875991 
```

Figure 15: Standard deviations

4.2.2 Graphical analysis

Histograms show distribution of continuous variables. Height of bars maps number of observations that fall into given bin.

```
histogram_plot <- function(dt, col) {
  ggplot(dt, aes(x = dt[, col])) +
```

```
    geom_histogram(aes(y = ..density..)) +
    labs(x = col) +
    stat_function(fun = dnorm, color = "red",
                  args = list(mean = mean(dt[[col]]),
                              sd = sd(dt[[col]])))
}

grid.arrange(
  histogram_plot(bikesharing_hourly, "temp"),

  histogram_plot(bikesharing_hourly, "atemp"),

  histogram_plot(bikesharing_hourly, "hum"),

  histogram_plot(bikesharing_hourly, "windspeed"),

  histogram_plot(bikesharing_hourly, "cnt"),

  nrow = 3
)
```

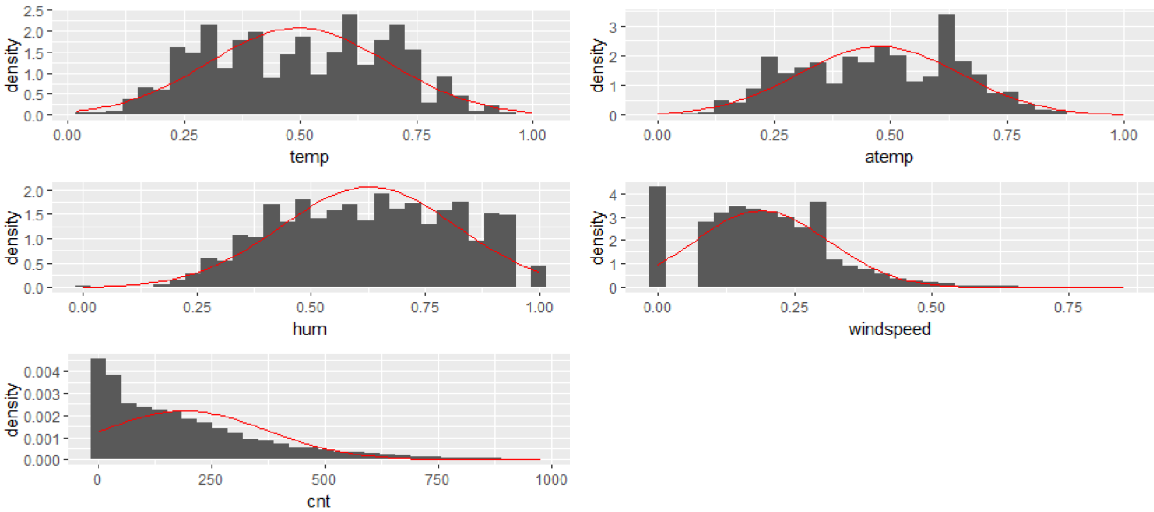

Figure 16: Histograms

Windspeed and count data are right-skewed which implies mean is higher than median. Boxplot is based around outliers as well as five-number summary statistics: minimum, 1st quartile, median, 3rd quartile and maximum. Whiskers contain almost all of the data, while its middle half falls inside the box and remainder is represented by points. Height of box shows IQR (interquartile range).

```r
grid.arrange(
  ggplot(bikesharing_hourly, aes(y = temp)) +
    geom_boxplot(),

  ggplot(bikesharing_hourly, aes(y = atemp)) +
    geom_boxplot(),

  ggplot(bikesharing_hourly, aes(y = hum)) +
    geom_boxplot(),

  ggplot(bikesharing_hourly, aes(y = windspeed)) +
    geom_boxplot(),

  ggplot(bikesharing_hourly, aes(y = cnt)) +
    geom_boxplot(),

  nrow = 2
)
```

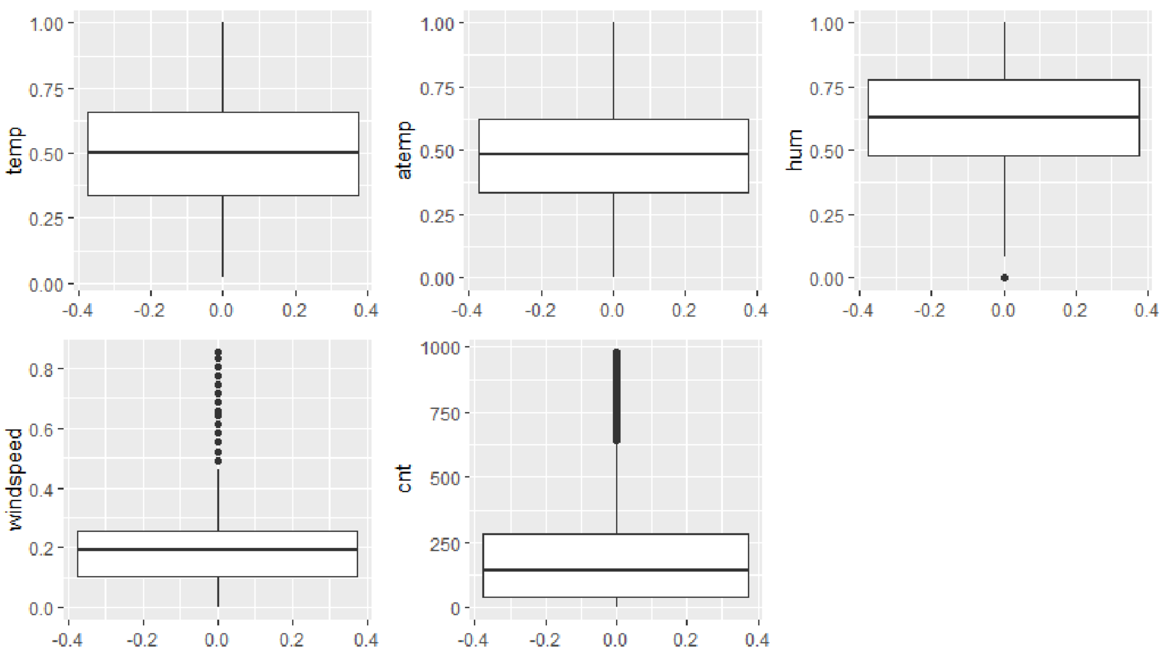

Figure 17: Boxplots

Outliers in wind speed and count of total rental bikes data may strongly affect mean and standard deviation. These are observations that have extreme values far from distribution scale. Interquartile range of cnt boxplot demonstrates high dispersion in count of total rental bikes data.

To explore how the count of total bike rentals depends on predictors, it will be visualised using scatter plots.

```
grid.arrange(
  ggplot(bikesharing_hourly, aes(x = season, y = cnt)) +
    geom_jitter(alpha = 0.2, shape = 20),

  ggplot(bikesharing_hourly, aes(x = yr, y = cnt)) +
    geom_jitter(alpha = 0.2, shape = 20),

  ggplot(bikesharing_hourly, aes(x = holiday, y = cnt)) +
    geom_jitter(alpha = 0.2, shape = 20),

  ggplot(bikesharing_hourly, aes(x = workingday, y = cnt)) +
    geom_jitter(alpha = 0.2, shape = 20),

  ggplot(bikesharing_hourly, aes(x = weathersit, y = cnt)) +
    geom_jitter(alpha = 0.2, shape = 20),

  nrow = 3
)
```

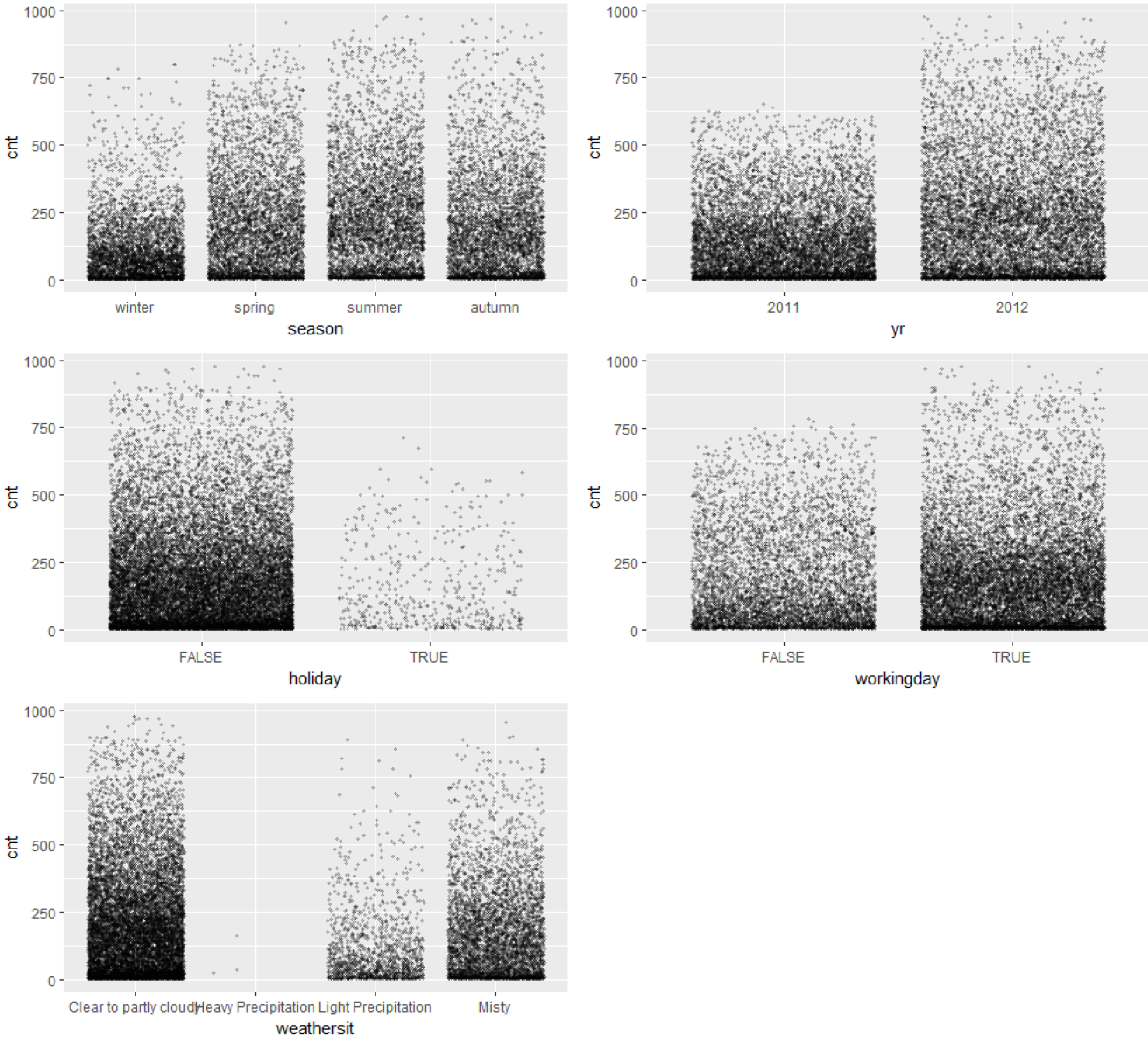

Figure 18: Scatter plots for season, year, holiday, working day, weather situation

Linear trend can be spotted in dependency of number of bike rentals on year, working day (positive) and holiday, weather situation (negative). Level of bike riders goes up from year 2011 to 2012 which may imply growth of bikesharing system's popularity. Users tend to commute by bicycle during working days and rent it less frequently during unfavourable weather conditions.

```
grid.arrange(
  ggplot(bikesharing_hourly, aes(x = mnth, y = cnt)) +
    geom_jitter(alpha = 0.2, shape = 20),

  ggplot(bikesharing_hourly, aes(x = hr, y = cnt)) +
    geom_jitter(alpha = 0.2, shape = 20),

```

```
  ggplot(bikesharing_hourly, aes(x = weekday, y = cnt)) +
    geom_jitter(alpha = 0.2, shape = 20)
)
```

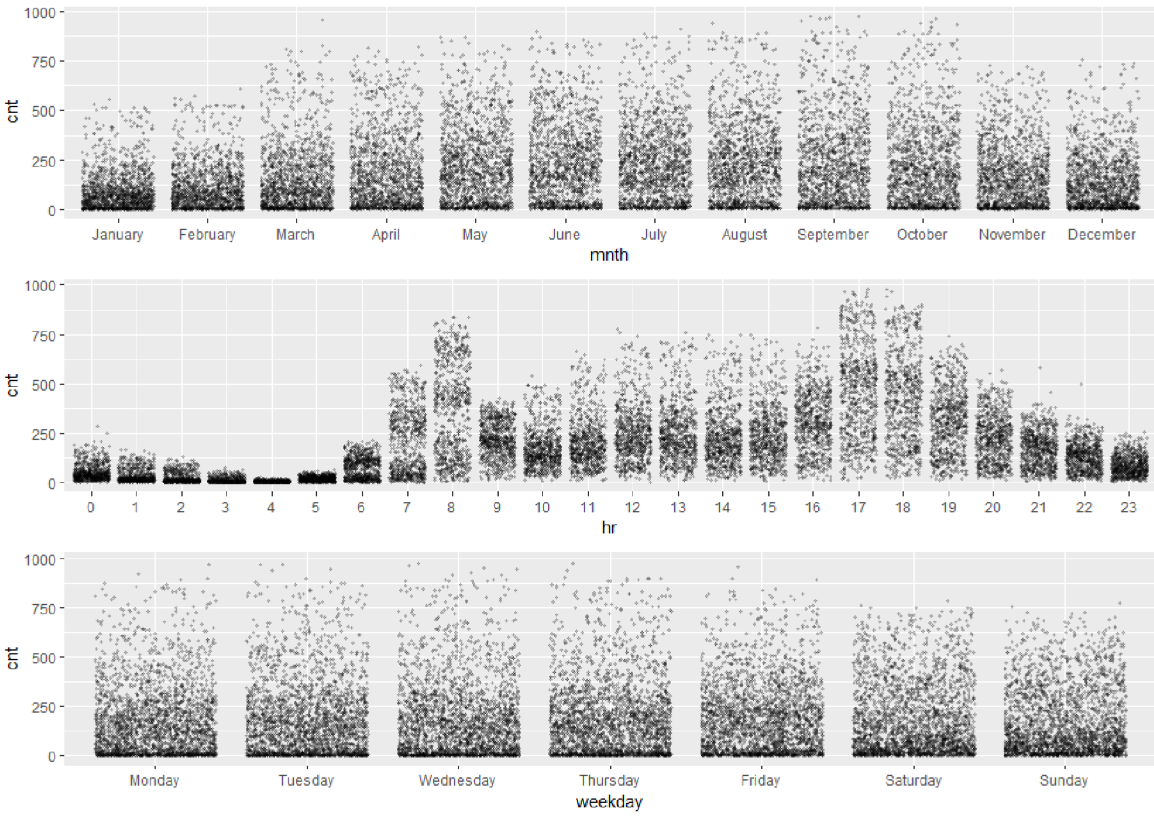

Figure 19: Scatter plots for month, hour, weekday

The peak in number of bike rentals falls in summer months of the year, whereas minimal use is observed in January (interrelated to season feature). At the beginning and end of the year, it increases and decreases accordingly. The lowest level of use is between 4 am and 5 am, and highest is at the time of rush hours (8 am and 5 - 6 pm). Weekday is significantly impacted by working day feature.

```
grid.arrange(
  ggplot(bikesharing_hourly, aes(x = temp, y = cnt)) +
    geom_count(alpha = 0.2, shape = 20),

  ggplot(bikesharing_hourly, aes(x = atemp, y = cnt)) +
    geom_count(alpha = 0.2, shape = 20),

  ggplot(bikesharing_hourly, aes(x = hum, y = cnt)) +
```

```
    geom_count(alpha = 0.2, shape = 20),

  ggplot(bikesharing_hourly, aes(x = windspeed, y = cnt)) +
    geom_count(alpha = 0.2, shape = 20),

  nrow = 2
)
```

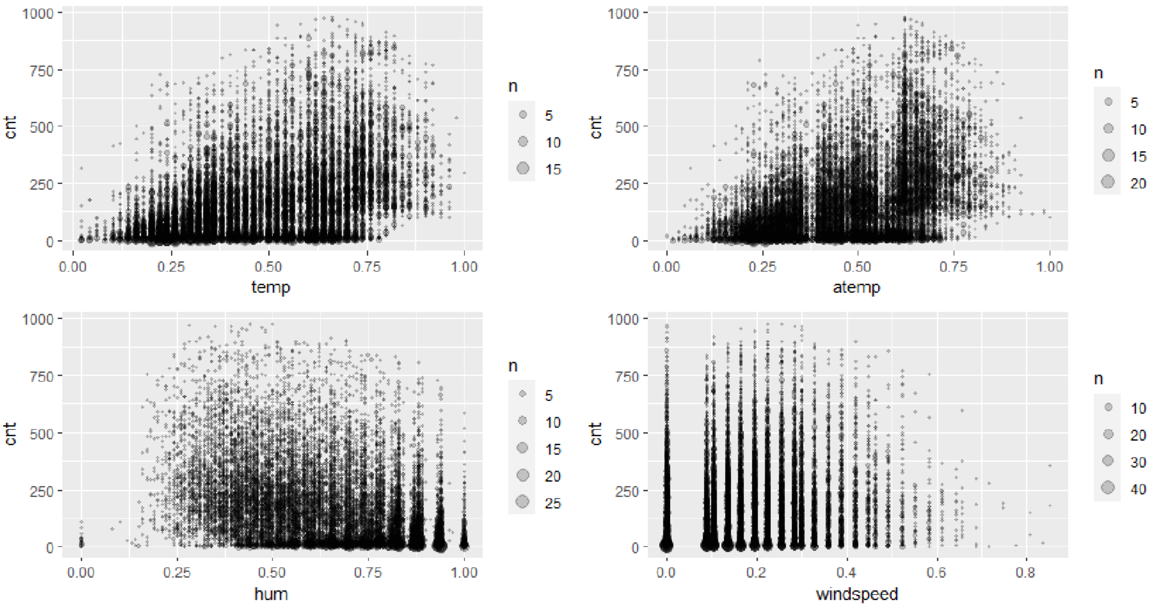

Figure 20: Scatter plots for temperature, feeling temperature, humidity, wind speed

There is a positive trend in the number of rental bikes to temperature and feeling temperature with drop at high temperatures, as people do not ride a bicycle when it is cold and extremely hot outside. Negative trend is observed in dependency of total count of users on humidity, wind speed. High level of humidity associated with higher chances of precipitation and strong wind indicate less rental of bicycles.

4.2.3 Task specification

Each observation is associated with a response value. The problem of predicting a real-valued variable by a set of predictors, is referred to as the regression problem. If features (year, month, hour, holiday, weekday, working day, weather situation, temperature, humidity, wind speed) are taken as predictors and number of bicycles rented per hour as response, then the original task falls under the definition of regression problem. Generally, it assumes trend, that is, describes the relationship between the count of total bike rentals and all other features.

4.2.4 Correlation analysis

Correlation matrix presents information on pairwise correlations that are calculated to check multicollinearity in explanatory variables.

```r
corr_matrix <- cor(x = bikesharing_hourly[, c(3:14)], method = "
    pearson")
corrplot(corr_matrix, method = "number", col = brewer.pal(n = 10, name
      = "RdBu"), tl.col="black")
```

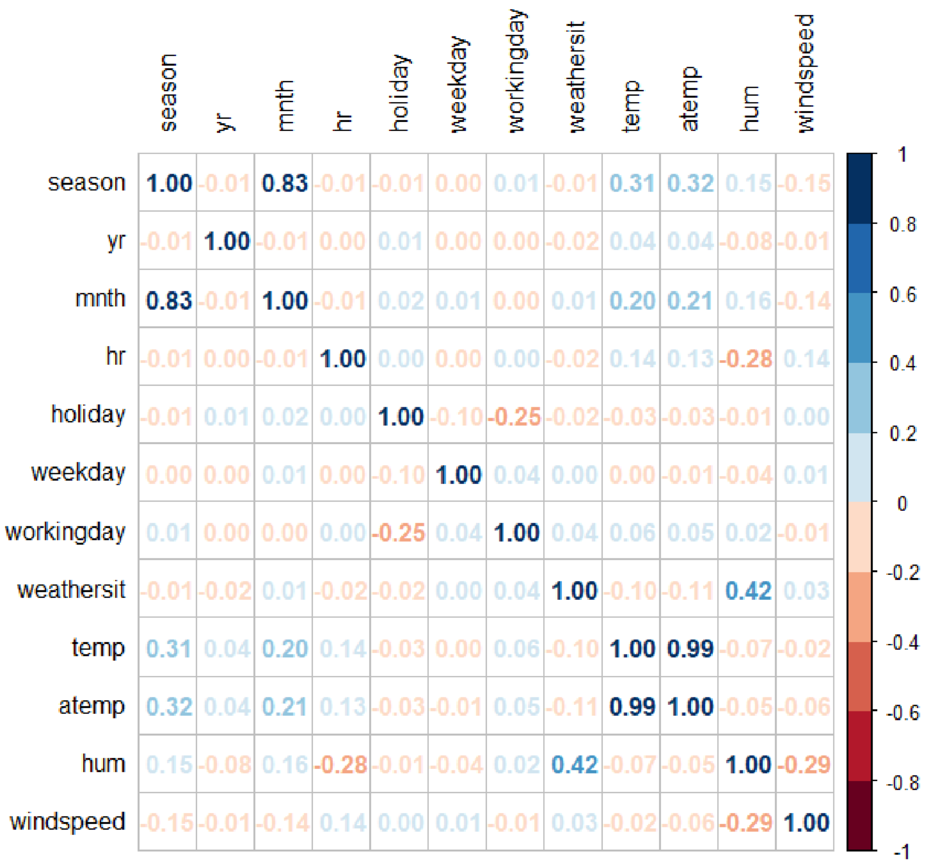

Figure 21: Correlogram

On the diagonal of correlogram, as intended, are "1" values. However, there are some features which are similar to each other. One can easily observe multicollinearity in two pairs of matrix columns – season and month, temperature and feeling temperature – coefficients of correlation are higher than 75%. Given that both are correlated by their nature, season and feeling temperature will be omitted from the model.

Thus, all the necessary data analysis has been done to proceed with construction of machine learning models.

4.3 Regression model

4.3.1 Train-Test Split

Seeing that dataset is time sensitive, it is splitted sequentially into training and test sets:

```
frac <- round(nrow(bikesharing_hourly) * 0.7)
train_set <- bikesharing_hourly[1:frac, ]
test_set <- bikesharing_hourly[(frac + 1):nrow(bikesharing_hourly), ]
```

First 70% rows of data are selected as train_set, while remaining 30%, respectively, as test_set.

4.3.2 Training

Mean and variance are estimated to determine regression (Poisson or Quasipoisson) to use for prediction of total rental bikes.

```
train_set %>%
  summarise(mean = mean(cnt), variance = var(cnt))
```

```
    mean variance
159.9337 23280.49
```

Figure 22: Mean and variance of count

Test for excessive variance is performed using `qcc.overdispersion.test`:

```
qcc.overdispersion.test(train_set$cnt, type = "poisson")
```

```
Overdispersion test Obs.Var/Theor.Var Statistic p-value
       poisson data          145.5634   1770634       0
```

Figure 23: Overdispersion test for count

On the basis of above output, variance is much different from mean. Assumption of mean and variance equality is not confirmed, therefore Quasipoisson regression will be applied. In supervised machine learning problem, unknown model parameters are evaluated by training set. Hence, model is learning on training data:

```r
reg_model <- glm(formula = cnt ~ yr + mnth + hr + holiday + weekday +
    workingday + weathersit + temp + hum + windspeed,
     data = train_set,
     family = quasipoisson)
```

```
Call:
glm(formula = cnt ~ yr + mnth + hr + holiday + weekday + workingday +
    weathersit + temp + hum + windspeed, family = quasipoisson,
    data = train_set)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-24.826   -3.497   -0.745    2.784   21.919

Coefficients: (1 not defined because of singularities)
                              Estimate Std. Error t value Pr(>|t|)
(Intercept)                   2.857130   0.044851  63.702  < 2e-16 ***
yr2012                        0.579949   0.010855  53.429  < 2e-16 ***
mnthFebruary                  0.110755   0.019986   5.542 3.06e-08 ***
mnthMarch                     0.300499   0.019816  15.164  < 2e-16 ***
mnthApril                     0.422721   0.020728  20.394  < 2e-16 ***
mnthMay                       0.464386   0.023800  19.512  < 2e-16 ***
mnthJune                      0.513450   0.030614  16.772  < 2e-16 ***
mnthJuly                      0.366235   0.032854  11.147  < 2e-16 ***
mnthAugust                    0.447431   0.030993  14.436  < 2e-16 ***
mnthSeptember                 0.591132   0.028563  20.696  < 2e-16 ***
mnthOctober                   0.675633   0.025046  26.976  < 2e-16 ***
mnthNovember                  0.593817   0.024518  24.220  < 2e-16 ***
mnthDecember                  0.482673   0.024516  19.688  < 2e-16 ***
hr1                          -0.454837   0.056166  -8.098 6.12e-16 ***
hr2                          -0.805792   0.063680 -12.654  < 2e-16 ***
hr3                          -1.449954   0.082394 -17.598  < 2e-16 ***
hr4                          -2.119306   0.110899 -19.110  < 2e-16 ***
hr5                          -0.996326   0.068850 -14.471  < 2e-16 ***
hr6                           0.398683   0.045708   8.722  < 2e-16 ***
hr7                           1.422482   0.039093  36.387  < 2e-16 ***
hr8                           1.947998   0.037325  52.190  < 2e-16 ***
hr9                           1.431043   0.038819  36.864  < 2e-16 ***
hr10                          1.141211   0.039952  28.565  < 2e-16 ***
hr11                          1.284323   0.039267  32.707  < 2e-16 ***
hr12                          1.460474   0.038703  37.736  < 2e-16 ***
hr13                          1.451730   0.038819  37.398  < 2e-16 ***
hr14                          1.386102   0.039116  35.436  < 2e-16 ***
hr15                          1.421325   0.039037  36.410  < 2e-16 ***
hr16                          1.633868   0.038377  42.574  < 2e-16 ***
hr17                          2.048423   0.037379  54.801  < 2e-16 ***
hr18                          1.988070   0.037390  53.172  < 2e-16 ***
hr19                          1.686927   0.037939  44.464  < 2e-16 ***
hr20                          1.376291   0.038868  35.410  < 2e-16 ***
hr21                          1.134125   0.039878  28.440  < 2e-16 ***
hr22                          0.881993   0.041243  21.385  < 2e-16 ***
hr23                          0.482166   0.044217  10.905  < 2e-16 ***
holidayTRUE                  -0.161224   0.026940  -5.985 2.23e-09 ***
weekdayTuesday                0.013292   0.014673   0.906   0.3650
weekdayWednesday             -0.003708   0.014887  -0.249   0.8033
weekdayThursday               0.027608   0.014571   1.895   0.0581 .
weekdayFriday                 0.057911   0.014479   4.000 6.38e-05 ***
weekdaySaturday               0.037375   0.014667   2.548   0.0108 *
weekdaySunday                -0.023969   0.014907  -1.608   0.1079
workingdayTRUE                      NA         NA      NA       NA
weathersitHeavy Precipitation -0.474762  0.353888  -1.342   0.1798
weathersitLight Precipitation -0.550652  0.019027 -28.941  < 2e-16 ***
weathersitMisty              -0.079605   0.009810  -8.114 5.35e-16 ***
temp                          1.216033   0.045469  26.744  < 2e-16 ***
hum                          -0.162916   0.026682  -6.106 1.05e-09 ***
windspeed                    -0.244353   0.032371  -7.548 4.72e-14 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for quasipoisson family taken to be 27.81735)

    Null deviance: 1695892  on 12164  degrees of freedom
Residual deviance:  345647  on 12116  degrees of freedom
AIC: NA

Number of Fisher Scoring iterations: 5
```

Figure 24: Regression model

Significance of each the independent variable is evaluated at p-value less than 0.05. Coefficient for workingday is identified as singular and will be removed from the model.

New model is constructed for training:

```
new_reg_model <- glm(formula = cnt ~ yr + mnth + hr + holiday +
    weekday + weathersit + temp + hum + windspeed,
                data = train_set,
                family = quasipoisson)
```

4.3.3 Prediction

The main task of machine learning techniques is to perform well on new data. Prediction for model is made on test set:

```
test_set$prediction <- predict(reg_model, newdata  = test_set, type =
    "response")
```

Graph is displayed to see how predicted values are comparable with actual counts.

```
ggplot(test_set, aes(x = prediction, y = cnt)) +
  geom_point(alpha = 0.2, shape = 20) +
  geom_abline(color = "darkblue")
```

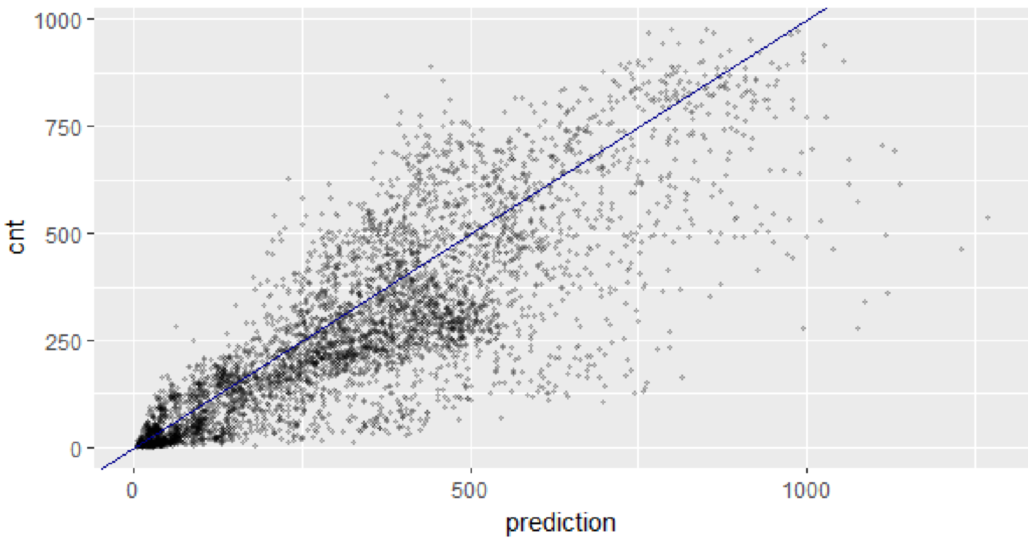

Figure 25: Regression model - predicted vs actual count

4.4 Neural network model

4.4.1 Normalisation

Obtain original data for humidity and windspeed to scale back from preliminary processing; in this way, all features proceed from the same form.

```
bikesharing_hourly <- bikesharing_hourly %>%
  mutate(hum = hum * 100,
    windspeed = windspeed * 67)
```

MinMax normalisation to [0; 1] scale is conducted for all features except temperature and feeling temperature seeing that they were normalised beforehand.

```
maxs <- apply(bikesharing_hourly[, c(3:10, 13:17)], 2, max)
mins <- apply(bikesharing_hourly[, c(3:10, 13:17)], 2, min)
scaled <- as.data.frame(scale(bikesharing_hourly[, c(3:10, 13:17)],
   center = mins, scale = maxs - mins))  %>%
  mutate(
    temp = bikesharing_hourly$temp,
    atemp = bikesharing_hourly$atemp)
```

4.4.2 Train-Test Split

Dataset is splitted into training and test sets in an orderly manner:

```
frac <- round(nrow(scaled) * 0.7)
train_set <- scaled[1:frac, ]
test_set <- scaled[(frac + 1):nrow(scaled), ]
```

This results in two samples of following sizes: scaled training set – 12,165 observations, scaled test set – 5,214 observations.

4.4.3 Training

`neuralnet` library is installed to solve count of bike rentals task, which accepts different arguments including model fit formula, data frame, number of hidden neurons for each layer, threshold.

Model of neural network is learning on training data with architecture of nine neurons in input layer, two hidden layers – first layer with four neurons and second layer with two neurons, and one neuron in output layer. Since regression problem is solved, a linear activation function $f(x) = x$ is used on output layer.

```
nn_model <- neuralnet(formula = cnt ~ yr + mnth + hr + holiday +
    weekday + weathersit + temp + hum + windspeed,
                      data = train_set,
                      hidden = c(4, 2),
                      threshold = 0.01,
                      linear.output = TRUE)
```

plot method is applied to view oriented graph of neural network, in which nodes are neurons and edges are connections between neurons with their weights. It consists of a sequence of layers, each contains a number of neurons.

```
plot(nn_model)
```

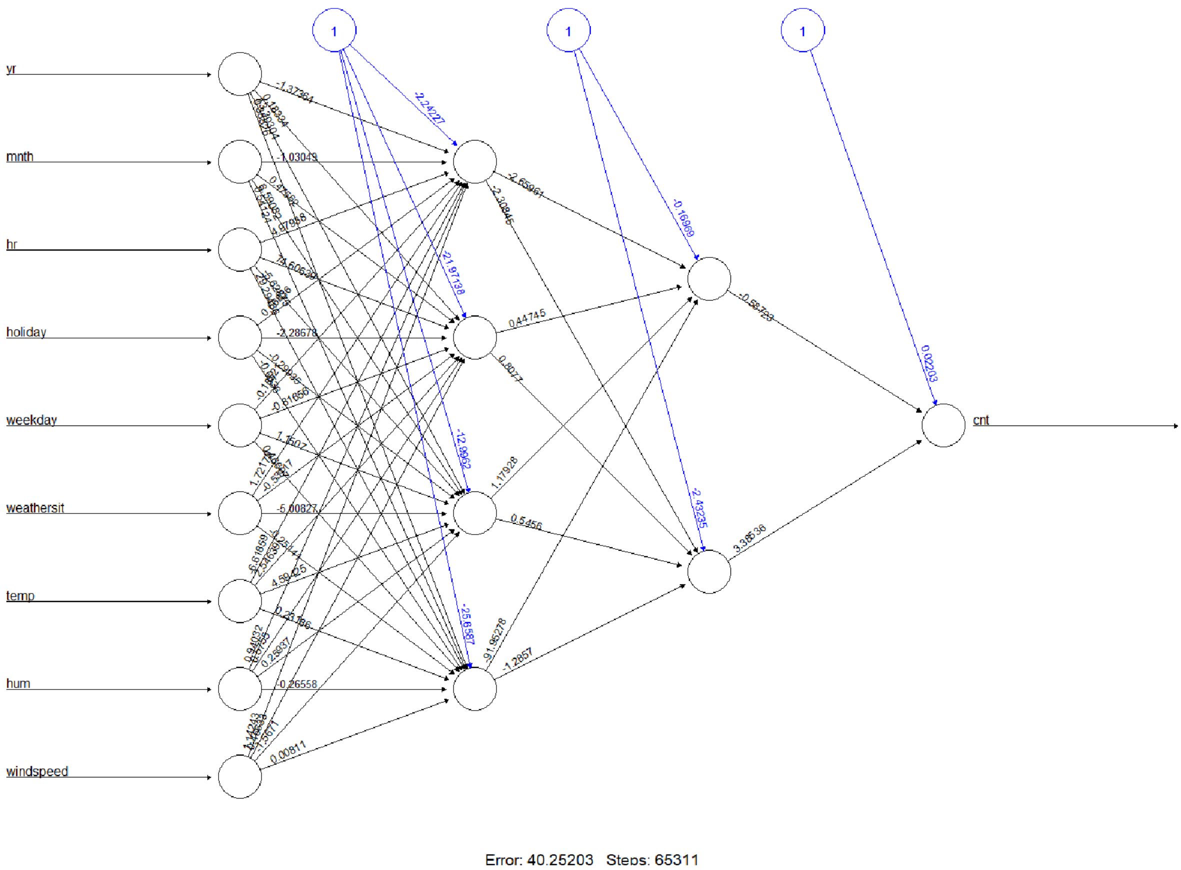

Figure 26: Neural network visualisation

Model error, number of taken steps, weights (black lines) and biases (blue lines) are retrieved by `result.matrix` attribute and displayed below.

```
nn_model$result.matrix
```

```
                                  [,1]
error                     4.025203e+01
reached.threshold         9.475528e-03
steps                     6.531100e+04
Intercept.to.1layhid1    -2.242268e+00
yr.to.1layhid1           -1.373640e+00
mnth.to.1layhid1         -1.030486e+00
hr.to.1layhid1            4.979379e+00
holiday.to.1layhid1       6.683577e-01
weekday.to.1layhid1      -1.852130e-01
weathersit.to.1layhid1    1.721792e+00
temp.to.1layhid1         -6.618587e+00
hum.to.1layhid1           9.403198e-01
windspeed.to.1layhid1     1.142427e+00
Intercept.to.1layhid2    -2.197138e+01
yr.to.1layhid2            1.833400e-01
mnth.to.1layhid2          4.756225e-01
hr.to.1layhid2            7.460639e+01
holiday.to.1layhid2      -2.286776e+00
weekday.to.1layhid2      -8.165561e-01
weathersit.to.1layhid2   -5.311661e-01
temp.to.1layhid2          2.546390e+00
hum.to.1layhid2           5.754983e-01
windspeed.to.1layhid2     4.663318e-01
Intercept.to.1layhid3    -1.299620e+01
yr.to.1layhid3            1.330304e+01
mnth.to.1layhid3          6.590816e+00
hr.to.1layhid3           -5.628747e+00
holiday.to.1layhid3      -2.993491e-01
weekday.to.1layhid3       1.150697e+00
weathersit.to.1layhid3   -5.008274e+00
temp.to.1layhid3          4.594252e+00
hum.to.1layhid3           2.593708e-01
windspeed.to.1layhid3    -1.567105e+00
Intercept.to.1layhid4    -2.565870e+01
yr.to.1layhid4            3.882590e-01
mnth.to.1layhid4          5.412393e-01
hr.to.1layhid4            2.929466e+01
holiday.to.1layhid4      -5.360035e-02
weekday.to.1layhid4       4.660733e-01
weathersit.to.1layhid4   -2.514142e-01
temp.to.1layhid4          2.318622e-01
hum.to.1layhid4          -2.655753e-01
windspeed.to.1layhid4     8.111407e-03
Intercept.to.2layhid1    -1.696851e-01
1layhid1.to.2layhid1     -2.659610e+00
1layhid2.to.2layhid1      4.474536e-01
1layhid3.to.2layhid1      1.179277e+00
1layhid4.to.2layhid1     -9.195278e+01
Intercept.to.2layhid2    -2.432351e+00
1layhid1.to.2layhid2     -2.308449e+00
1layhid2.to.2layhid2      8.077049e-01
1layhid3.to.2layhid2      5.456021e-01
1layhid4.to.2layhid2     -1.285697e+00
Intercept.to.cnt          2.203275e-02
2layhid1.to.cnt          -5.872343e-01
2layhid2.to.cnt           3.385355e+00
```

Figure 27: Resulting matrix of neural network model

4.4.4 Prediction

After training, prediction of target variable is made on test data.

```
nn_prediction <- data.frame(prediction = predict(nn_model, newdata =
    test_set))
```

```
test_set$prediction <- nn_prediction$prediction
```

Values of variables were normalised based on MinMax method. Consequently, `scale_back` function has been created, which is defined as follows:

```
scale_back <- function(x, y) {
  x * (max(y) - min(y)) + min(y)
}
```

Above mentioned function is then applied to convert predicted and actual count of bikes rented in an hour.

```
test_ <- data.frame(prediction = scale_back(test_set$prediction, bikesharing_hourly$cnt),
                    cnt = scale_back(test_set$cnt, bikesharing_hourly$cnt))
}
```

Following outcome is obtained by plotting graph to make a comparison between predicted and actual count.

```
ggplot(test_,
       aes(x = prediction, y = cnt)) +
  geom_point(alpha = 0.2, shape = 20) +
  geom_abline(color = "darkblue")
```

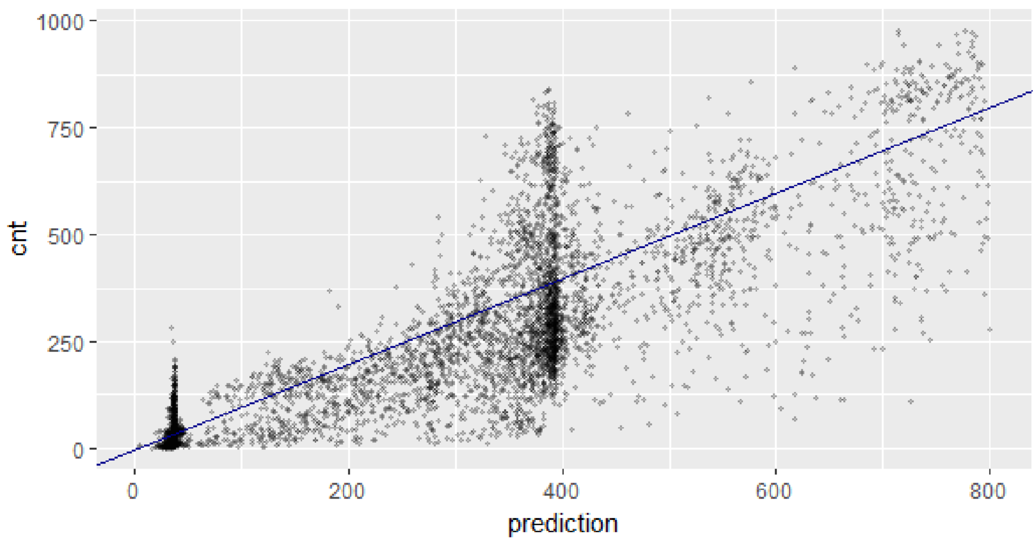

Figure 28: Neural network model - predicted vs actual count

5 Results and Discussion

After the prediction, performance of models in bikesharing demand task is assessed using Mean Absolute Error, Mean Squared Error and Root Mean Square Error. These measurements are part of the `Metrics` package.

MAE

Mean Absolute Error is used to evaluate the quality of model. The table 29 provides results of regression and neural network models assessment:

```
data.table(Model = c("Regression", "Neural network"),
           MAE = c(mae(test_set$cnt, test_set$prediction),
                   mae(test_$cnt, test_$prediction)))
```

```
            Model      MAE
1:     Regression 94.16826
2: Neural network 97.66415
```

Figure 29: Table of MAE values

By the value of MAE quality measurement, better performance turned out to be in regression model with error of 94.

MSE

As can be seen from the data table 30, neural network model performed better on a scaled sample.

```
data.table(Model = c("Regression", "Neural network"),
           MSE = c(mse(test_set$cnt, test_set$prediction),
                   mse(test_$cnt, test_$prediction)))
```

```
            Model      MSE
1:     Regression 18624.24
2: Neural network 17759.99
```

Figure 30: Table of MSE values

RMSE

Results of this metric are presented in figure 31:

```
data.table(Model = c("Regression", "Neural network"),
           RMSE = c(rmse(test_set$cnt, test_set$prediction),
                    rmse(test_$cnt, test_$prediction)))
```

	Model	RMSE
1:	Regression	136.4707
2:	Neural network	133.2666

Figure 31: Table of RMSE values

As shown above, neural network model has made slightly smaller prediction error than regression model.

In the process of machine learning techniques assessment for predicting count of total rental bikes, following findings has been deduced. Regression model demonstrated that it:

- is not sensitive to feature scaling,
- does not require meticulous configuration of parameters,
- handles various data types equally well without additional processing.

On the flip side, neural network architecture demands more rigorous parameters tuning and data processing:

- requires normalisation of features to the uniform scale,
- computationally demanding – training ran over 65,000 steps which took approximately eight minutes of calculation for a graph of sixteen nodes.

```
36000      min thresh: 0.0186698423878443
37000      min thresh: 0.0186698423878443
38000      min thresh: 0.0176259597644912
39000      min thresh: 0.0170095318474467
40000      min thresh: 0.0158055886245168
41000      min thresh: 0.0151913419263541
42000      min thresh: 0.0145800105792902
43000      min thresh: 0.0145150409385625
44000      min thresh: 0.0138566919212934
45000      min thresh: 0.0132622426334254
46000      min thresh: 0.0128079475883319
47000      min thresh: 0.0128079475883319
48000      min thresh: 0.0124279769646658
49000      min thresh: 0.0123908966774003
50000      min thresh: 0.0115640075958952
51000      min thresh: 0.0115640075958952
52000      min thresh: 0.0115640075958952
53000      min thresh: 0.0115640075958952
54000      min thresh: 0.0113658788739438
55000      min thresh: 0.0105466196353883
56000      min thresh: 0.0105466196353883
57000      min thresh: 0.0105466196353883
58000      min thresh: 0.0103790149551222
59000      min thresh: 0.0103790149551222
60000      min thresh: 0.0103790149551222
61000      min thresh: 0.0103790149551222
62000      min thresh: 0.0103790149551222
63000      min thresh: 0.0103790149551222
64000      min thresh: 0.0100094557329354
65000      min thresh: 0.0100094557329354
65311      error: 40.25203 time: 8.87 mins
```

Figure 32: Excerpt from neural network training

As a follow-up to this work, it is possible to formulate and solve the task with multiple variations in the number of hidden layer(s) and neurons within them in neural network considering that various configurations can show different performance and might improve the overall quality of the model.

6 Conclusion

In the framework of this work, two techniques of supervised machine learning were reviewed and applied to the task of bikesharing demand prediction. Classification and regression problems of supervised learning were also discussed. Scientific literature was analyzed and knowledge about the subject was formed. All necessary data preprocessing and detailed analysis were conducted. Models were constructed to features such as count of total bike rentals, year, month, hour, holiday, weekday, working day, weather situation, temperature, humidity and wind speed. The results include graphs comparing actual and projected count values and a tables of performance evaluations.

It can be concluded that the use of neural network and regression models may be appropriate in the task of bikesharing demand prediction. Neural network model achieved better results in terms of Mean Squared Error and Root Mean Square Error. It is also worth noting that regression model showed higher quality in Mean Absolute Error and a bit behind in Root-Mean-Square Error in comparison with neural network. On a final note, it has been summarised that regression model can be quite significant to predict number of rented bikes, and therefore, there are grounds to employ it to this task.

Bibliography

AKALIN, Altuna. 2020. *Computational Genomics with R.* CRC Press.

AMEISEN, Emmanuel. 2020. *Building Machine Learning Powered Applications: Going from Idea to Product.* Sebastopol: O'Reilly Media, Inc.

ANDERSEN, Lars Bo, SCHNOHR, Peter, SCHROLL, Marianne, & HEIN, Hans Ole. 2000. All-cause mortality associated with physical activity during leisure time, work, sports, and cycling to work. *Archives of internal medicine*, **160**(11).

ANDERSON, Edgar. 1935. The irises of the Gaspe Peninsula. *Bulletin American Iris Society*, **59**.

BASHA, Syed Muzamil; RAJPUT, Dharmendra Singh. 2019. *Survey on evaluating the performance of machine learning algorithms: past contributions and future roadmap.* Academic Press.

BENGFORT, Benjamin, BILBRO, Rebecca, & OJEDA, Tony. 2018. *Applied text analysis with Python: Enabling language-aware data products with machine learning.* Sebastopol: O'Reilly Media, Inc.

BLACK, C.; POTTER, S. 1999. Bikeabout in Portsmouth: Ein automatisiertes Fahrradverleihsystem mit Smart-Card. SEMACO GES.M.B.H.

BRINK, Henrik, RICHARDS, Joseph, & FETHEROLF, Mark. 2016. *Real-world machine learning.* Shelter Island: Manning Publications Co.

BÜHRMANN, Sebastian. 2007. *New Seamless Mobility Services: Public Bicycles.* NICHES Policy Notes.

CAVILL, Nick; DAVIS, Adrian. 2007. *Cycling and health-what's the evidence?* London: Cycling England.

CHOLLET, Francois. 2021. *Deep learning with Python.* Manning Publications.

DATE, Sachin. 2019. *The Poisson Regression Model.* [Online] September 22. [Cited: December, 2022.] `https://timeseriesreasoning.com/contents/poisson-regression-model/`.

DEMAIO, Paul. 2009. Bike-sharing: History, impacts, models of provision, and future. *Journal of public transportation*, **12**(4).

DEMAIO, Paul, O'BRIEN, Oliver, RABELLO, Renata, CHOU, Steve, & BENICCHIO, Thiago. 2023. *The Meddin Bike-sharing World Map.* [Online] February 26. [Cited: January, 2023.] `https://bikesharingworldmap.com/index.php`.

EL HAMI, Abdelkhalak; POUGNET, Philippe (ed). 2020. *Embedded Mechatronic Systems 2: Analysis of Failures, Modeling, Simulation and Optimization.* Elsevier.

FANAEE-T, Hadi. 2013. *Bike Sharing Dataset - UC Irvine Machine Learning Repository.* [Cited: December, 2022.] `https://archive-beta.ics.uci.edu/dataset/275/bike+sharing+dataset`.

FERREIRA FILHO, Wladston; PICTET, Raimondo. 2021. *Computer Science Unleashed: Harness the Power of Computational Systems.* Las Vegas: Code Energy.

Google Developers. 2022. *Supervised Learning — Machine Learning — Google Developers.* [Online] July 18. [Cited: December, 2022.] `https://developers.google.com/machine-learning/intro-to-ml/supervised`.

GRUS, Joel. 2019. *Data Science from Scratch: First Principles with Python.* Sebastopol: O'Reilly Media.

HENLEY, John. 2005. *Rentabike moves up a gear from curiosity to runaway success.* [Online] August 12. [Cited: December, 2022.] `https://www.theguardian.com/world/2005/aug/12/france.jonhenley`.

HURBANS, Rishal. 2020. *Grokking artificial intelligence algorithms.* Shelter Island: Manning Publications.

IANSITI, Marco; LAKHANI, Karim R. 2020. *Competing in the age of AI: Strategy and leadership when algorithms and networks run the world.* Boston: Harvard Business Press.

KABACOFF, Robert. 2022. *R in Action: Data Analysis and Graphics with R and Tidyverse.* Simon and Schuster.

KAMBEZIDIS, H. D.; SAYIGH, A. 2012. *Comprehensive Renewable Energy.* Elsevier.

LANTZ, Brett. 2019. *Machine learning with R: expert techniques for predictive modeling.* Birmingham: Packt Publishing Ltd.

LEA, Perry. 2018. *Internet of Things for Architects: Architecting IoT solutions by implementing sensors, communication infrastructure, edge computing, analytics, and security.* Birmingham: Packt Publishing Ltd.

LONGLEY, James W. 1967. An appraisal of least squares programs for the electronic computer from the point of view of the user. *Journal of the American Statistical association*, **62**(319).

MA, C. C. Y.; IQBAL, M. 1984. Statistical comparison of solar radiation correlations. *Solar Energy*, **33**(2).

MORDENSKY, Stanley Paul, LIPOR, John J, BURNS, Erick R., & LINDSEY, Cary R. 2022. What did they just say? Building a Rosetta stone for geoscience and machine learning.

MÜLLER, Andreas C.; GUIDO, Sarah. 2016. *Introduction to machine learning with Python: a guide for data scientists*. Sebastopol: O'Reilly Media, Inc.

NADAL, Luc. 2007. Bike sharing sweeps Paris off its feet. *Sustainable Transport*.

NG, Annalyn; SOO, Kenneth. 2017. *Numsense! Data Science for the Layman: No Math Added*. Annalyn Ng and Kenneth Soo.

NIELSEN, Birgitte Høy. 1993. *The Bicycle in Denmark: Present Use and Future Potential*. Reading, Massachusetts: Ministry of Transport.

OBIS. 2009. *Optimising Bike Sharing in European Cities: A Handbook*. Tech. rept.

O'SULLIVAN, Feargus. 2022. *The Dutch Anarchists Who Launched a Bike-sharing Revolution - Bloomberg*. [Online] February 26. [Cited: December, 2022.] `https://www.bloomberg.com/news/features/2022-02-26/the-dutch-anarchists-who-launched-a-bikesharing-revolution`.

PBSC Urban Solutions. 2021. *The Meddin Bike-sharing World Map: Mid-2021 Report*. Tech. rept.

PBSC Urban Solutions. 2022. *The Meddin Bike-sharing World Map Report: 2022 edition*. Tech. rept.

RANI, Alka, KUMAR, Nirmal, KUMAR, Jitendra, & SINHA, Nishant K. 2022. *Machine learning for soil moisture assessment*. Academic Press.

RASCHKA, Sebastian; MIRJALILI, Vahid. 2019. *Python machine learning: Machine learning and deep learning with Python, scikit-learn, and TensorFlow 2*. Birmingham: Packt Publishing Ltd.

ROBACK, Paul; LEGLER, Julie. 2021. *Beyond multiple linear regression: applied generalized linear models and multilevel models in R*. Boca Raton: CRC Press.

ROMERO, Carlos. 2008. SpiCycles in Barcelona.

SCHNEIDER, Patrick; XHAFA, Fatos. 2022. *Anomaly Detection and Complex Event Processing Over IoT Data Streams: With Application to EHealth and Patient Data Monitoring*. Academic Press.

SCHUTT, Rachel; O'NEIL, Cathy. 2014. *Doing data science: Straight talk from the frontline*. Sebastopol: O'Reilly.

SHAHEEN, Susan; COHEN, Adam P. 2014. *Bicycle Sharing*. SAGE Publications.

SHEPHARD, Roy J. 2008. Is active commuting the answer to population health? *Sports medicine*, **39**(9).

SHUKLA, Nishant; FRICKLAS, Kenneth. 2018. *Machine learning with TensorFlow*. Manning.

SUBASI, Abdulhamit. 2022. *Applications of Artificial Intelligence in Medical Imaging*. Academic Press.

TRASK, Andrew W. 2019. *Grokking deep learning*. Manning Publications.

VENKATESWARLU, Ch; KARRI, Rama Rao. 2022. *Optimal State Estimation for Process Monitoring, Fault Diagnosis and Control*. Elsevier.

WOODFORD, Chris. 2021. *How neural networks work - A simple introduction*. [Online] August 30. [Cited: December, 2022.] `https://www.explainthatstuff.com/introduction-to-neural-networks.html`.