Hedonic Pricing in the Horse Market: Inspection of Multimodal Features as Price Determinants in a Peer-to-peer Luxury Goods E-commerce

Author: Michele Sergio Pozzi
Supervisor: dr. Anastasija Tetereva
Second assessor: Vardan Avagyan

School of Economics
Erasmus University

A thesis submitted for the degree of:
Master of Economics and Business

Programme name:
Data Science and Marketing Analytics

Rotterdam 2020

Abstract

The horse market is rather peculiar. Horses are unique, luxury goods. Moreover, with the advent of e-commerce, horse trading has moved to online marketplaces. These e-commerce platforms allow international and peer-to-peer trades. This thesis uses a hedonic pricing approach to better understand the dynamics characterising this complex market. The developed models fully exploit the variety of retrievable data regarding animals for sale on e-commerce websites. In fact, numeric, categorical, text and image features are collected from online ads. Advanced machine learning models are created to better process the different data types. Through blackbox model-agnostic methods, the effect of the different features on the price is assessed. The results have several marketing implications. They restitute valuable insights regarding product development, promotion and pricing. For instance, it is found that the parts of a horse's picture which depict its front have a positive impact on the predicted price. Thus, images showing the face of the horse should be used to better promote the animal.

Keywords: hedonic pricing, horse market, multimodal fusion regression, blackbox model-agnostic methods.

Contents

1 Introduction

Fusaichi Pegasus is a rather intriguing name. It mixes an Asian sound with the image of the mythological winged horse. Indeed, this is the name of a legendary stud. Fusaichi Pegasus holds the record of the most expensive horse ever sold. In 2000, it was purchased for 60 million US dollars (Hunter, 2003). This is an exceptional case, but it cannot be denied that horses are luxury goods. In fact, the costs to purchase and maintain a horse are high on average. According to the data collected for the research, the average price of a horse is around $6,000. Moreover, keeping a horse costs about $4,000 per year ("Guide to First-Time Horse Ownership," n.d.). However, the high costs do not seem to stop the equine industry from thriving. It is estimated that the equine industry is accountable for a turnover of 300 billion dollars worldwide (Equine Business Association, n.d.). The actors involved in this big industry are diverse. On the one hand, people with all kinds of experience are interested in buying a horse to participate in equestrian sports. In the last years, there has been an increase in the amount of amateur riders with a limited knowledge of horse's husbandry (Gille et al., 2010). On the other hand, horses are sold by breeders or by sport practitioners who decided to change their ride or to stop riding. Moreover, both sellers and buyers might be interested or specialised in only one of the many equestrian sports (e.g. racing, jumping, dressage). The advent of e-commerce websites has furnished platforms on which this variety of actors can get in contact with each other. Many specialised websites have been created to support the trading of horses among peers (Freeborn, 2009). These websites also allow connecting demand and offer coming from different countries.

Given the described context, correctly pricing a horse is important but challenging. It is important because it is one of the business actions at the base of the functioning of the equestrian industry. Moreover, it usually leads to notable investments by the involved player. At the same time, it is challenging because, as said, the goods to be priced are luxury goods, most of the traders lack expertise in horse breeding, most of the buyers purchase a limited amount of horses throughout their lives, the actors negotiate on a peer-to-peer basis and the goods might be traded internationally. In addition to this, it must be considered that horses are unique goods, they differ by nature in genotype and phenotype.

One way to address difficult pricing decisions is to compare similar products based on a set of characteristics in order to formulate an educated guess of their value. This practice is also called hedonic pricing. Luckily, on e-commerce platforms for horse trading, ads contain numerous pieces of information about the animal. For the present research images, text, categorical and numerical data are collected from online ads. These data can be called "multimodal", i.e. they differ in the modality or way in which they happen and are experienced (Baltrušaitis et al., 2018). The present study proposes to investigate how to build a hedonic pricing model for determining the price of horses, which includes all the multimodal data at disposal. Thus, the research question is:

"To what extent do the multimodal features of a horse's ad on an online marketplace determine its price, when considering a deep learning approach?"

To answer the research question, four hedonic pricing models are designed. The first three accept data of only one modality as input, i.e. either images, text, or the other variables. Thus, they are said "unimodal". This gives three advantages. First, it enables us to use

modality-specific regression techniques, e.g. *convolutional neural networks* for images (CNN). Second, it quickly assesses how well each data modality can determine the price of a horse. Indeed, it is argued that the better a unimodal model performs, the more informative is the data modality used by it. Third, unimodal models allow using modality-specific *model-agnostic techniques*. These are methods to assess which features, among the ones of a certain modality, are most relevant and how they influence the model predictions. The fourth model takes all the available data as input. It is named the *multimodal fusion model*. It allows us to answer the research question further by assessing the extent to which the price can be determined by the combination of all the data at hand. It is found that the unimodal model for text data outperforms the one for numeric and categorical variables. The unimodal model for images performs worse than the other two. The multimodal model, in turn, outperforms the unimodal ones. Furthermore, model-agnostic techniques applied to unimodal models give very interesting results. Many of the numeric, categorical, text, and image features are found to be relevant price determinants and their effect on the price is unveiled as well.

This thesis gives a relevant contribution to academic literature. This is one of the few studies using multimodal data for pricing (see subsection 2.4 for more). Moreover, this is the first study considering such a variety of data to build a hedonic pricing model for the horse market. Furthermore, to analyse these complex data, this thesis uses *neural networks* (NN). It is rare to see such sophisticated models used in hedonic pricing. While they generally lead to an improvement of performances, they are more difficult to interpret compared to other regression techniques, such as linear regression. To exploit the power of NNs and still be able to interpret the results, model-agnostic techniques are employed. Thus, this thesis proposes more advanced techniques to solve hedonic pricing models, as well as contributes to the research field of model-agnostic methods, which is young, even if rapidly growing. Finally, this study develops a multimodal fusion model specific to the problem at hand. Creating multimodal fusion models is never trivial and there is no standard way to do so. Thus, this thesis contributes also to this branch of literature.

The results of this study have several implications for marketing practices in the horse industry. Indeed, it can be said that they touch three of the four famous marketing Ps: product, promotion and price (McCarthy, 1964). In fact, the unimodal models for numeric or categorical variables and for text data reveal which breeds, genders, horse's skills and coat colours lead to higher or lower prices. This information can be used by horse breeders to breed and train animals which are more in line with buyers' expectations, i.e. develop better products. Furthermore, the characteristics of the ads, of the text and of the images which have a positive effect on the predicted price are uncovered. This teaches the sellers what kind of communication approach maximises the perceived value of the product. Additionally, based on these results, the website would be able to develop a tool to suggest the people that post ads which words and images can be used to optimise their promotion strategy. Furthermore, the models developed can help to effectively solve horse pricing decisions. For instance, the multimodal model could be used to develop a website tool suggesting the correct price to the users, thus helping to effectively clear the market. Outside of the horse industry, the results of this thesis prove that a similar approach could be employed in similar contexts. Other peer-to-peer online marketplaces that are specialised in a good could take advantage of such models, even for what regards luxury goods trade.

2 Related works

This thesis brings together a classical piece of economic theory and some of the most advanced data science techniques. The hedonic pricing theory was first formulated in the sixties. It has already been applied several times to different business contexts, including the horse trade. However, most of the hedonic pricing models use simple linear regressions. This thesis, instead, employs advanced machine learning methods to obtain better results, to analyse image data, and to exploit multimodal data. Thus, the first part of this chapter explains where this piece of economic theory lays its basis, why it is relevant for marketing purposes, and how it was used in the horse trading context. The second subsection, instead, gives an introduction to some of the youngest fields of studies in machine learning. Moreover, it summarises the few papers which use these machine learning techniques to tackle similar pricing problems.

2.1 Hedonic pricing, marketing analysis and the horse market

This thesis proposes to create hedonic pricing models. These models attempt to predict the price of an object based on its characteristics. They are based on Lancaster's (1966) approach to consumer theory, which states that the price of a good is determined by its characteristics. Already in this first paper, it is highlighted how this new approach to consumer theory could help marketing analysts in developing, pricing and advertising new products (Lancaster, 1966). Rosen (1974) first introduces the term "hedonic prices" and further elaborates Lancaster's (1966) theories. Hedonic prices are defined as the implicit prices of products' characteristics. Rosen (1974) highlights how hedonic prices are strictly related to the buyers' and sellers' decision making process. Since its invention, hedonic pricing has been extensively applied in the real-estate context, as shown in a literature review made by Sirmans et al. (2005). In many of the reviewed papers, hedonic pricing is used to unveil the effectiveness of marketing strategies. Furthermore, there is a plethora of papers not regarding the real estate market, in which hedonic pricing is shown to have relevant implications for marketing analysis (Donnet et al., 2007; Hopkinson & Pujari, 1999; Steiner, 2004). Thus, previous literature shows that the use of hedonic pricing models leads to many marketing implications. Indeed, it can be used to guide not only pricing, but also promotion and product development decisions. Therefore, hedonic pricing studies are related to three of the four marketing Ps: price, product and promotion (McCarthy, 1964). This refers to the famous framework which summarises the key elements of marketing to four basic principles all starting with "p" (price, product, promotion and placement).

There have been several attempts to create hedonic pricing models for the horse trade. In the past, hedonic pricing analysis has been done for race-bred yearling quarter horses (Lansford et al., 1998), show quality quarter horses (Taylor et al., 2006), thoroughbred broodmares in foal (Maynard & Stoeppel, 2007) and Texan ranch horses (Lange et al., 2010). All these studies are limited in the number, the breed and the type of horses analysed. Moreover, the horses considered were all sold at auctions. A study which considers web-scraped data from an online marketplace for horse trade was written more recently (Freeborn, 2009). In this paper, the best performing model reached an R-squared of 33%. However, unstructured data were not used in this analysis. An attempt to extract valuable insights from text found in horse advertisements focused on understanding whether Australian riders valued safety when

buying a horse (Oddie et al., 2014). This last paper does not give any evaluation of the model accuracy. So far, no study has included images in hedonic price analysis for horse trade, let alone both images and text. Moreover, until now, all the hedonic pricing models for horse trade have used linear regressions.

2.3 Machine learning and pricing

To include all the available data in the hedonic pricing models, this study relies on machine learning methods. Indeed, it touches three different machine learning research domains: computer vision, multimodal fusion and blackbox model-agnostic methods. There are several papers related to the present study coming from each of these research fields.

To start with, computer vision is the branch of study which deals with processing image and video data with a computer (Joo & Steinert-Threlkeld, 2018). Convolutional neural-networks are among the most used computer vision techniques (Joo & Steinert-Threlkeld, 2018). The history of CNNs is linked with the history of the *Imagenet* dataset. Imagenet is a large database composed of several millions of images labelled through crowd sourcing (Deng et al., 2009). This database has been used as a benchmark to evaluate the effectiveness of new CNN architectures proposed by researchers. The creation of deep learning models which achieve good performances on Imagenet is of great improtance. Indeed, it has been proven that the CNNs which perform well on Imagenet will also achieve better performances when trained on other tasks (Kornblith et al., 2019). Among the famous CNN architectures which were benchmarked on Imagenet, there are, in chronological order, *AlexNet*, *VGGnet*, *Inception*, *ResNet* and *DenseNet* (Alom et al., 2018). Each of the mentioned models outperformed the previous ones in terms of classification accuracy. Researchers have also put effort into designing CNN architectures with a contained number of parameters and a small model size, which were still able to perform decently on Imagenet. For instance, *SqueezeNet* (Iandola et al., 2016) and *MobileNets* (Howard et al., 2017) are two famous fruits of these efforts. In 2019, Tan and Le have presented a family of CNNs named *EfficientNets*, which are both efficient and effective. Indeed, EfficientNets achieve state-of-the-art performances, while having less parameters and being faster than most CNNs. An example of computer vision used in a context similar to the one of this thesis is given by Chen et al. (2018). This paper demonstrates that a convolutional neural network can predict the price of a bicycle or a car by only taking its raw image as an input. Some ways to perform model-agnostic methods on the trained CNNs are discussed as well. This proves that images could be used to determine the price of a product and that it could be understood which parts of a picture trigger a certain prediction.

However, the data at hand are composed not only by images, but also by text, numeric and categorical variables. Hence, the data differ in the way or modality in which they are encountered: they are multimodal. CNNs are not suited to process multimodal data. Other artificial intelligence methods are usually applied to solve multimodal problems, i.e. problems which deal with multimodal data. Baltrušaitis et al. (2018) provide a taxonomy and an extensive summary of multimodal machine learning applications. In their paper, they use the term multimodal fusion to indicate machine learning techniques which use multimodal data to make a prediction. A review of literature on multimodal fusion for multimedia analysis is given by Atrey et al. (2010). Neural networks are often used in multimodal fusion models (Baltrušaitis et al., 2018). An example of how multimodal fusion can be used to tackle pricing

tasks is given by Ahmed and Moustafa (2016). In this paper, researchers have demonstrated that using visual features in addition to the usual numerical variables improves the quality of the estimation of house prices (Ahmed & Moustafa, 2016). This is relevant for our research because it shows that visual features are relevant price determinants even for houses which, as horses, are highly expensive goods. However, in contrast to what is done in this thesis, at first, Ahmed and Moustafa (2016) use features extractors to preprocess multimodal data and, only in a second moment, they run a regression on the extracted features vectors. Moreover, they do not provide any explanation of how visual features influence the predictions. An approach to multimodal regression which better resembles the proposed one is found in Ortega et al. (2019). In this paper, a deep neural network architecture is proposed to perform an audio-video sentiment scoring task. However, the relation between the thesis and this last paper stops at the similarity in the proposed model. In fact, the tasks to which the models are applied are completely different.

Finally, computer vision and multimodal fusion models have proven to be extremely useful in performing tasks which more traditional techniques could not tackle efficiently. However, these extremely complex models are more challenging to interpret than, for instance, traditional linear regressions. The exploration of model-agnostic methods is an active area of research. Blackbox model-agnostic techniques is just a jargon term for methods which interpret a machine learning model. Molnar (2019) gives a review of the most basic and widespread model-agnostic techniques. Moreover, as previously mentioned, Chen et al. (2018) describe and apply four different CNN's model-agnostic techniques: *saliency maps*, *gradient-weighted class activation maps*, *local interpretable model-agnostic explanations* and *sliding window heatmaps*. Only the latter can be applied to regression models and, unfortunately, it only gives back local model explanations. A local explanation shows how the model performs one particular prediction, not how the algorithm works in general. Luckily, theoretical and practical guidelines on how to obtain a global interpretation of a model from local explanations are given by Ribeiro et al. (2016). Their paper is also extremely interesting for other reasons. In fact, on the one hand, it contains conceptual requirements that a good model-agnostic technique shall meet, on the other hand, it introduces a new way to explain classifiers' predictions. Finally, no research has been done on the interpretation of multimodal fusion networks (Baltrušaitis et al., 2018). Model-agnostic methods are of great importance for the present paper. Indeed, this thesis attempts not only to predict the price of a horse, but also to understand which characteristics of the data lead to a certain prediction. Thus, the papers mentioned above are very relevant.

2.6 Final comments on related works

This thesis contributes to the previous works in many ways. First, it takes data into consideration which have previously been ignored in hedonic pricing models for the horse market, such as images. Second, it further explores model-agnostic methods for computer vision algorithms, a field which lends itself for more research. Third, it attempts to use raw images as input in a multiple inputs model, while, in the previous literature on hedonic pricing, models considering multiple inputs first extracted features from the images and then inserted them as predictors into a second separate model.

3 Data

The data required for the present study were not found in any database that was already available online. Nobody had already conducted a similar research. Thus, it was decided to web-scrape the data needed. An accurate selection procedure was designed to decide which website to download the data from, also following the guidelines of Freeborn (2009). *Equinenow* was found to best suit the research requirements. To better understand why *Equinenow* was chosen over the many other horse-trade websites, see Appendix A.

3.1 Functioning of Equinenow

Equinenow hosts ads regarding horses for sale mainly in Canada or the US. All the advertisements are displayed on the website for a time span of three months. Most ads include data describing the horse (e.g. height and breed), a picture and a lengthy text description in English. For most of the horses a price is given, but, when an animal is sold, the price is blocked out and only a red label saying "Sold" is displayed.

Equinenow allows the seller to post 4 different types of ads:

- basic text ad
- standard photo ad
- premium ad
- premium plus ad.

All the ads are displayed on the website for three months. The first two types of ads are free. Premium ads cost $14.95 and premium plus ads cost $24.95. Basic text ads display some info about the horse (e.g. height, breed) and a textual description of it. In standard photo ads also one picture can be displayed. Premium ads allow the seller to post as many as 8 pictures and to embed a video. In premium plus ads even more pictures (15) can be displayed. The website claims to show premium plus ads above premium ads, premium ads above standard photo ads and standard photo ads above basic text ads. Moreover, by paying a fee, any add can get the "featured" status, which gives additional visibility to the ad by, for instance, randomly showing it in the sidebar or in the top bar of the website.

3.2 Data description

On the 28th of March 2020, data were harvested from *Equinenow*'s ads in which a price was displayed. Data from 6729 ads were downloaded, of these ads only 6544 had an image. Every ad is uniquely identified by a number (*adNumber*). Furthermore, a text description is provided for every horse. In addition to the price, many other numerical and categorical variables are collected for every ad. In total, every ad in the created database is described by 33 different variables, of which four are textual, and by an image. In the following subsections, a description of the numeric and categorical variables, textual data and image data will be given.

3.2.1 Numerical and categorical variables

There are 29 numeric or categorical variables that describe every ad. For every ad, eight pieces of information are always present on the website. Among these, there is the *adNumber*, a number that uniquely identifies the ad. Moreover, the price (*price*) is always given. 341 prices are expressed in Canadian dollars, 42 prices in euros, one in Polish zloty and the remaining 6345 are in US dollars. All the prices are converted to USD using the current exchange rate taken from www.xe.com. A variable called *currency* is created to store the original currency of the price. An entry with a price higher than 2 billion is considered a mistake and dropped. The same happened for a horse costing $200,300. The maximum price is $200,000, the minimum is $3, the average price is $5,848, the median is $3,500 and the standard deviation is $7,810. Figure 1 shows the distribution of the prices lower than $50,000.

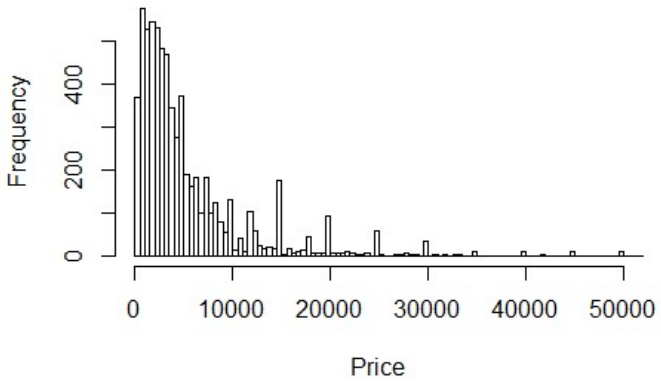

Figure 1: Distribution of prices lower than 50,000 dollars.

A third piece of information available on the website for every ad is the location of the seller. The location is given either as a name of a state (i.e. Virginia, Alberta) or as a city name followed by a state abbreviation (i.e. "Rising Sun, MD"). In the latter case, the name of the state is extracted from the state code while the name of the city is dropped, so that all the location data can be accessed in a standard form, i.e. the state name. The horses sold on the website come from 70 different states. This piece of information is saved in the column *location*. The gender of the horse is always given as well (*Gender*). There are 11 different gender descriptions, the most common ones are "Gelding" and "Mare". It is also indicated if the horse is in foal, i.e. pregnant (*InFoal*). This happens to be true in only 131 cases. A score of the horse's temperament is reported. The disposition of the horse is scored on a decimal scale, with 1 indicating a gentle horse, 10 an aggressive one. The score is assigned by the seller, the average score is of 4.45. This score is saved as *Temperament*. Finally, the dates in which an ad was created, and in which it was updated for the last time were downloaded and stored (*AdCreated*, *LastUpdate*).

More data describing the horses was retrieved from the website and organised in 12 variables. In these variables, some missing values (NA) are present. The variable *Breed* accounts for the breed of the specimen. There are 90 different breeds, the most represented one is "Quarter

Horse" (1779). The only missing value value was set to be "Unknown". The state in which the horse was bred (*StateBred*) is reported as well. This information is missing for almost half of the data points. All the NAs are replaced by "Not Given". The colour of a horse (*Color*) can be of forty different shades, "Bay" is the most common one. The 1485 missing values are set to be "Unknown". *Markings* describes if a horse coat has got any markings. 1215 different markings are attributed to 2261 horses. NA values are substituted with "None". Height and weight (*Height*, *Weight*) are reported respectively in hands (hh) and pounds (lbs). Height values which were lower than 4 hh and weight values which were lower than 57 lbs were set to be NAs, since those were the measures of Thumbelina, the smallest horse that ever existed (Douglas, 2007). The missing values in *Height* are 1259 in total, in *Weight* 1748. The foal date, i.e. the date of birth, is another important feature (*FoalDate*). *FoalDate'*s which appear to be in the future are set to NAs, since they cannot be realistic. The same reasoning goes for the dates preceding 1958, since Old Billy, the oldest horse that ever lived, died at age 62 (Meier, 2013). The dates are then converted to the age of the horse in years. The oldest horse in the database is 36, the youngest are new-born foals and the average age is around 8. Overall, there are 2863 missing values in *FoalDate*. To fill in the missing values, a first attempt to automatically infer the age of the horse from the textual information contained in the ad is done. Indeed, often the foal date or the age of the horse is reported in the text. This way, 2005 missing values could be filled up. Going back to the variables' description, some ads indicate to which breed registry a horse is registered (*Registry*). A horse can be registered into a breed registry only after experts have tested its pedigree. 110 different registries are present in the data. There are 3590 missing values, which are transformed to "No Registration". The registration number is provided for 1400 horses, in the other ads this information is missing (*RegistryNumber*). The name, saved as *name*, of the specimen is present in most of the ads (4758). The name of the farm in which the animal was bred is present in 3317 ads (*ownerName*). Finally, a description of the skills of the horse and of the disciplines in which it was trained is present in 70% of the ads. Every horse can have multiple skills and disciplines, they are reported in alphabetical order and separated by commas. 242 skills and disciplines are mentioned in the *skillsDisciplines* column including "Rodeo", "Beginner" and "Champion".

The 858 NAs present in *FoalDate,* the 1259 present in *Height* and the 1748 in *Weight* were imputed using the missForest algorithm (Stekhoven & Bühlmann, 2012). This non-parametric algorithm leverages on the random forest (RF) technique to impute missing values in multiple columns of a dataset and accepts mixed-types data as input. It has been proven to outperform other imputation algorithms, such as k-nearest neighbours' imputation or multivariate imputation using chained equations (Stekhoven & Bühlmann, 2012). It functions by: first, filling in the NAs in a dataset by mean imputation; second, training a random forest model to predict one of the features with missing values; third, substituting the previous estimates of the NAs with the ones made by the RF and then repeating the second step but with another column as response variable (Stekhoven & Bühlmann, 2012). In this thesis, the missForest algorithm was used to fill in the NAs of *FoalDate*, *Height* and *Weight*. To impute the missing values other variables were used as predictors as well. These include: *location*, *Breed*, *StateBred*, *Gender*, *InFoal* and *Temperament*. The imputation error expressed as normalised root-mean-square error is 19.3%.

Some data could also be harvested about the features characterising the ads. *hasVideo* accounts for the presence of a video embedded in the ad, this was the case only for 332 advertisements. The variables *hasPicture*, *hasMultiplePictures* and *noOfPictures* report respectively if an ad has at least one picture, more than one picture and the total number of pictures included in it. 184 ads haven't got any picture, 722 have more than one photo and all the rest have only one. The type of ad is not explicitly reported on the webpage, but it could be surmised out of the last 4 variables mentioned. Most ads are standard photo ads. The ad type is saved in the column *typeOfAd*. Finally, *hasPedigree*, *hasShippingNotes* and *hasOwnerDescription* indicate the presence of a space dedicated to a horse pedigree, additional notes about the shipment of the specimen or additional description of the breeding farm. The pedigree of the horse is rarely reported, in roughly one sixth of the ads there are additional shipping notes and in half of them there is a description of the selling farm.

3.2.2 Text data

The website allows the seller to attach some text to the ad organised in four different sections: a brief description, some additional comments, shipping notes and a description of the seller. The brief description is given for every horse. It consists of a few words summarising the main characteristics of the horse. It is placed in the top of the page and it is displayed besides the animal's name in the browsing page. It contains on average slightly more than 22 characters. An additional comment about the horse is always present. This is a lengthier text, on average of 665 characters, in which the specimen is described. It is placed at the bottom of the page. In the database, it is saved as *AdditionalComments*. A seller can add some notes about how the horse can be shipped to the buyer (*ShippingNotes*). This text, as already mentioned, is often missing. The shipping notes, when present, are displayed below the additional comments and are composed of an average of 80 characters. Finally, a description of the farm selling the horse can be attached at the bottom of the page (*OwnerDescription*). On average, it has 282 characters and it is present in roughly half of the advertisements.

3.2.3 Images

In 97% of the harvested ads, at least one picture was displayed. The first picture attached to every ad was downloaded. In fact, the first picture in the gallery of an ad is the one displayed in the browsing window to the customers searching for horses. It is also displayed in a larger size when the ad's webpage is first opened. Thus, the first picture in the gallery is also the most relevant in conveying information about the horses. Moreover, in 86% of ads, only one picture is displayed.

All the images are saved in jpg format and named after the ad number. The downloaded images have different pixel ratios (i.e. shapes) and resolutions. Some ads happen to display the same picture. This happens often for ads posted by the same owner and with pictures in which more than one horse is shown.

3.3 Data conclusion and limitations

This is, to the best of our knowledge, the first database about horses at sale including categorical, numerical, textual and image data. It is one of few multimodal data collections to

train a hedonic pricing model. It contains 6727 data points, described by 33 columns and 6544 pictures. *Table 3* collects all the variables mentioned in the previous subsections. We put our efforts into creating this dataset also hoping that other academics will use it in the future.

It is worth noticing that the database has got some limitations. First, the prices are listed by the sellers. It is not known if the price will be met by the market demand and if, in the case of a sale, the transaction price will be equal to the listed price. However, more than 10,000 horses are sold on the website every three months, thus it seems like the demand often meets the offer. Second, as discussed in the previous subsections, many of the harvested data are imprecise or partially incomplete. This happens because it is up to the seller to fill in the ads' information and, sometimes, this is done poorly. Luckily, a holistic approach which uses text and images to estimate the prices should mitigate the impact of an inaccurate variable on the end result. Finally, all the rows in the database are unique, however a few ads which differ in the identification number are suspiciously similar under many, but not all, variables, and sometimes even in the picture. This could be due to an attempt by a seller to make their ad appear in as many queries as possible by slightly changing some of the variables in them. After all, placing an ad is free. This problem is extremely difficult to individuate and to address, since two similar ads might as well describe two similar specimens bred by the same farm and advertised with the same image. Hence, nothing is done to address this problem.

Table 1: Summary of variables at disposal

	Variable Name	**Type of variable**		**Variable Name**	**Type of variable**
1	*adNumber*	Categorical	**18**	*RegistryNumber*	Categorical
2	*price*	Numeric	**19**	*name*	Categorical
3	*currency*	Categorical	**20**	*ownerName*	Categorical
4	*Location*	Categorical	**21**	*skillsDisciplines*	Categorical
5	*Gender*	Categorical	**22**	*hasVideo*	Binary
6	*InFoal*	Binary	**23**	*hasPicture*	Binary
7	*Temperament*	Numeric	**24**	*hasMultiplePictures*	Binary
8	*AdCreated*	Ordinal	**25**	*noOfPictures*	Numeric
9	*LastUpdate*	Ordinal	**26**	*typeOfAd*	Categorical
10	*Breed*	Categorical	**27**	*hasPedigree,*	Binary
11	*StateBred*	Categorical	**28**	*hasShippingNotes*	Binary
12	*Color*	Categorical	**29**	*hasOwnerDescription*	Binary
13	*Markings*	Categorical	**30**	*smallDescription*	Text
14	*Height*	Numeric	**31**	*AdditionalComments*	Text
15	*Weight*	Numeric	**32**	*ShippingNotes*	Text
16	*FoalDate*	Numeric	**33**	*ownerDescription*	Text
17	*Registry*	Categorical			

4 Methods

The approach to the problem is developed based on one consideration: the data at hand are multimodal. In fact, images, text and the other variables differ in their modality. Indeed, images are experienced through vision, written text is an expression of natural language and numerical or categorical variables are the fruit of some sort of measurement. Therefore, at first, a unimodal approach to hedonic pricing is employed, and then a multimodal one. Once again, the term "unimodal" refers to a model which treats data of only one modality. On the contrary, a multimodal model deals with data of multiple modalities.

At first, three different unimodal hedonic pricing models are explored. Each model leverages on only one data type, i.e. either on images, on text or on the other variables. This grants many advantages. First, even if neural networks are used in all three cases, the network architecture which better suits the type of data is deployed, i.e. the convolutional architecture for the computer vision model. Furthermore, an evaluation of each of the models quickly shows to which extent the data types at hand are relevant for the price prediction. Finally, model-agnostic techniques specific for each of the built networks can be applied to understand which data features have the most relevant effect on the prediction outcome.

Secondly, a multimodal fusion model is created. As already mentioned in section 2.3, multimodal fusion models integrate data given in different modalities to perform a classification or regression task (Baltrušaitis et al., 2018). Neural networks have often been used for such models with successful results (Baltrušaitis et al., 2018). Indeed, previous research has shown that multimodal fusion neural networks are able to learn from big quantities of data and show good performances in comparison to other techniques (Baltrušaitis et al., 2018). Different network architectures have been proposed to solve numerous multimodal problems. In this case, the fourth model consists of a neural network with three different input branches, one for each data type, which are merged by a joint hidden layer. This architecture resembles others that were proposed in previous studies (Ortega et al., 2019; Rosebrock, 2019; Sun et al., 2017). This last model will be able to exploit the data coming from all the sources and the eventual presence of complementary information to improve the prediction accuracy. This is expected to lead to lower errors. However, there are no model-agnostic techniques to interpret such a complex model (Baltrušaitis et al., 2018).

While the input data vary a great deal among the models, the independent variable is the same. All the models attempt to predict the price of the horses. To be more precise, since the price distribution is exponential (see Figure 1), the logarithm of the price is taken as the independent variable. This way, the effect of the outliers over the loss function is smoothed out.

This thesis makes intense use of neural networks. Neural networks are very advanced statistic techniques (Goodfellow et al., 2016). They are said to be machine learning techniques because you can teach them how to solve problems. Like humans, they can learn from their mistakes, thanks to what is called *backpropagation* (Goodfellow et al., 2016). The learning process of neural networks is called model training or fitting (Goodfellow et al., 2016). Neural networks' behaviour depends on many parameters or hyperparameters. To have a good performing neural network, the parameters shall be optimized. The process of finding the best parameters is called hyperparameter tuning (Goodfellow et al., 2016).

Since neural networks tackle many different tasks, they can take various forms (Goodfellow et al., 2016). For instance, in this thesis, the first two models employ a very simple type of NNs (*deep feedforward neural networks*). The third model uses neural networks which can understand the content of pictures, the convolutional neural networks (Goodfellow et al., 2016). As mentioned, the fourth model is a neural network able to combine different inputs. Unfortunately, it is complex to understand how neural networks make decisions. This is why researchers invented new interpretation methods, called model-agnostic techniques (Molnar, 2019). These methods allow us to have a better understanding of how the NNs make decisions. There are different model-agnostic methods for different types of NNs. The more complex is the NN, the harder is to interpret it. For instance, we will not be able to give an interpretation of the fourth multimodal fusion model.

Neural networks are used in all four models for many reasons. First, they are the state of the art for multimodal and computer vision models (Joo & Steinert-Threlkeld, 2018). Second, NNs generally also reach good performances in text analytics and regression tasks. Finally, using models of the same class helps to compare the models' performances more fairly.

When neural networks are compared with other commonly used regression techniques, their usefulness for this study is even more evident. For instance, while most hedonic pricing models use linear regressions, neural networks are able to describe not only linear, but also nonlinear relationships between the predictors and the dependent variable (Goodfellow et al., 2016). Moreover, neural networks automatically detect existing interactions in the input data (Goodfellow et al., 2016). Not to mention that it would be impossible to feed images directly into a linear regression model. Still, linear regressions are easier to interpret than neural networks (Goodfellow et al., 2016). Thus, NNs shall be used only when they actually perform better than linear models. To check if this holds true, when possible, the neural networks' performances are compared to the ones of linear baseline models. Tree-based methods are also commonly used in machine learning applications. As neural networks, these methods are able to automatically detect interactions and to uncover nonlinearities. However, tree-based methods are as difficult to interpret as NNs. Moreover, they are not able to effectively mine information from image data. Therefore, these methods are not ideal to analyse multimodal data containing images.

Having outlined broadly and justified the proposed methodology, in the next subsections, more details about the methods are given. At first, an introduction to neural networks is provided. The following sections describe how every different hedonic pricing model is built and interpreted. Finally, model evaluation procedures are exposed.

4.1 General introduction to neural networks

Neural networks are advanced machine learning techniques. They are called "neural" because they somehow resemble the functioning of the neural nervous system (Goodfellow et al., 2016). They are called "networks" because they are composed of a network of interconnected perceptrons. Perceptrons are computational units which, as neurons, take multiple inputs and give a simple output (Goodfellow et al., 2016). The mathematical formula of a simple perceptron with m inputs is:

$$\varphi(w_0 + \sum_{i=1}^{m} x_i w_i))$$

where x_i is the ith input, φ is the activation function, w_0 is the bias term and $\sum_{i=1}^{m} x_i w_i$ is the weighted sum of the inputs. In other words, a perceptron computes a weighted sum of the input vector, adds a bias term, and then applies a function to it. The result of the perceptron is a single number. The weights W are the only part of the perceptron that needs to be trained.

All neural networks use simple perceptrons as basic building blocks for more complex architectures. Different architectures may vary a great deal. However, in general, all neural networks consist of a series of consecutive layers. Each layer is composed by multiple perceptrons, also called nodes. The nodes in one layer are only connected to the ones of the previous and of the following layer. The first layer of the network is fed with external data, it is called the input layer and it has got as many nodes as the number of input values (Friedman et al., 2001). The last layer of the network restitutes the result of the algorithm and it is called the output layer (Friedman et al., 2001). For regression problems, the output layer has only one node (Goodfellow et al., 2016). Between the input and the output layer, there could be some additional layers called hidden layers. The number of nodes in the hidden layers may vary. Usually, nonlinear activation functions are used in their nodes. Having nonlinear activation functions in the hidden layers enables the neural network to learn nonlinear problems, hence it is crucial for the good functioning of the model (Friedman et al., 2001).

However, this is only a broad description of how neural networks work. In this thesis, three different network architectures are used: deep feedforward neural networks are used for text and numeric and categorical variables, convolutional neural networks are used for image data and *multi-input neural networks* are used for the multimodal fusion model. A more detailed description of these architectures is given in the sequent subsections.

4.1.1 Introduction to deep feedforward neural networks

Deep feedfoward neural networks are the most widespread neural network architectures (Goodfellow et al., 2016). They perform well when applied to classification or regression tasks (Friedman et al., 2001). They take tabular data as input, i.e. data arranged in a simple table. They are able to uncover nonlinear relationships between the input and the output. Moreover, they allow for interactions between the input features.

Deep feedforward neural networks are called feedforward because they use forward propagation to generate results, i.e. the input layer passes data to the next layer and so on and so forth, until the output is given (Goodfellow et al., 2016). Moreover, they are said to be deep because they have at least one hidden layer, which gives depth to the model. They are composed of an input layer with as many nodes as the input values, some dense hidden layers and, in this case, a single node output layer with a linear activation function. The hidden layers are said to be dense because they are fully connected with the preceding layers, which means that every node of these layers receives the output of every node of the previous layer as input. This allows for interactions between the input data. The activation function of the hidden layers

must be nonlinear, as already mentioned in subsection 4.1. This enables the network to solve nonlinear problems. In this thesis, in the hidden layers, the *rectified linear unit* (ReLU) is used as the activation function. This function is recommended as default nonlinear activation for modern neural networks (Goodfellow et al., 2016). Its mathematical formula is:

$$\varphi(z) = max(0, z).$$

Thus, ReLU, outputs the input itself if the input is above 0. Elsewhere, it outputs 0.

Even if deep feedforward neural networks are ideal for processing tabular data, they fail to extract insights from images. Indeed, image data are different from tabular data, which is why a different architecture needs to be introduced for them.

4.1.2 Introduction to convolutional neural networks

Image data are rather peculiar. In fact, the computer sees an image as a three-dimensional matrix, where one of the dimensions is the height, another is the width and the third is the colour space of the image. For instance, an image of 224x224 pixels encoded in an RGB colour space is represented by a 224x224x3 matrix A. $A_{1,224,3}$ represents the intensity of the colour blue in the pixel in the top right corner of the image. The total number of entries in A is 150,528. Thus, images are highly dimensional data with 3-d spatial connotations. Moreover, the information included in an image are often redundant. For instance, the pieces of information contained in one pixel are also contained in the neighbouring ones.

All these characteristics make images a unique type of data. The network architectures which are better fit to deal with this kind of data are convolutional neural networks (Joo & Steinert-Threlkeld, 2018).

The convolutional neural networks take an image as input, under the form of a matrix. They extract features from the image, then they use these features to perform a regression or a classification task. They are generally composed of a few convolutional blocks, followed by an output head (Joo & Steinert-Threlkeld, 2018). The output head is nothing but a series of fully connected layers which transforms the data processed in the convolutional blocks into the final output (Joo & Steinert-Threlkeld, 2018). The convolutional blocks perform a series of operations which enable the network to extract features from an image. These blocks employ two operations which have not yet been discussed in this research: convolution and pooling (Goodfellow et al., 2016).

The convolution operation gives the name to CNNs (Goodfellow et al., 2016). The convolution is a linear operation which can be written as:

$$(f * g)(t) = \int f(x)g(t - x)dx.$$

f and g are two functions, and $*$ is the symbol for convolution (Goodfellow et al., 2016). $f(t)$ is also called the *input* and $g(t)$ the *kernel* (Goodfellow et al., 2016). If f and g are discrete functions, the convolution can be rewritten as:

$$(f * g)(t) = \sum_x f(x)g(t - x).$$

Then, the convolution becomes the sum of the multiplications of two functions where the input of one function is equal to the input of the other, but reversed and shifted by t.

In CNNs, the kernel is usually called *filter*. If, as in this thesis, the CNN takes as input an image represented by a three-dimensional matrix, the convolution function for a three-dimensional input is:

$$F(x, y, c') = \sum_{c=1}^{m} \sum_{i=1}^{h} \sum_{j=1}^{w} I(x + i - 1, y + j - 1, c) \cdot K(i, j, c, c').$$

F is called the *feature map*, it is a three-dimensional matrix where the output of the convolution is stored, and $F(x, y, c')$ is an entry of this matrix. The input I is the three-dimensional matrix representing the input data. $I(x, y, c)$ is the entry at position (x, y, c) of the matrix I. The filter K is a four-dimensional array of dimensions $w \times h \times m \times n$, where m is the depth of I and n is the desired depth of the matrix output by the convolutional layer (Joo & Steinert-Threlkeld, 2018). $K(x, y, c, c')$ is the entry at position (x, y, c, c') of the matrix K. More intuitively, the convolution function multiplies the pixel values of the pixel neighbouring the pixel in position (x, y) by a set of predefined parameters in the kernel, sums up the multiplications and gives back a number which synthetises the information of the pixels in a certain region of the image.

In a convolutional layer, the convolution operation is repeated many times using different values of x and y, thus on different locations of the picture data and for all the n different sets of parameters of the kernel (Joo & Steinert-Threlkeld, 2018). The results of such operations are stored in the feature map at the coordinates of the respective x, y and n.

Convolutional layers grant many advantages. First, they apply the same operation to every subsection of the input data, so they can learn a recurrent pattern independently from its position in the picture (Joo & Steinert-Threlkeld, 2018). Secondly, convolutional layers allow for weight sharing, i.e. applying the same kernel to every part of the input map (Joo & Steinert-Threlkeld, 2018). This means that only the parameters of K need to be fitted, hence the number of weights to train is drastically reduced (Joo & Steinert-Threlkeld, 2018). Third, they ensure local and sparse connectivity. Indeed, the kernel connects the output of one layer to only a few of the inputs of the layer (Joo & Steinert-Threlkeld, 2018). This is even more true in light of the fact that most of the kernels have a height and width of less than 10 (Joo & Steinert-Threlkeld, 2018).

In a convolutional block, the convolutional layer is followed by an activation layer. In the activation layer, the output of the convolutional layer is passed through a nonlinear activation function, usually ReLU is used (Joo & Steinert-Threlkeld, 2018).

The last layer of the convolutional block is a pooling layer. The pooling layer takes as input a feature map and gives as output another feature map with lower dimensions (Joo & Steinert-Threlkeld, 2018). It does so by pooling together neighbouring values thanks to an aggregating statistic. In practice, the feature map is divided in spatial neighbours called pooling windows, then an aggregating statistic is computed over the values in every window (Joo & Steinert-Threlkeld, 2018). Eventually, only one value per window is produced. For example, max pooling means that the maximum value in the window is returned, average

pooling, instead, gives back an average of the values in the window. The pooling layer reduces the dimensions of the data; hence it reduces the number of parameters to train. It does so by eliminating data redundancy (Joo & Steinert-Threlkeld, 2018). Moreover, it reduces the effect of small variations in the data over the model output, i.e. it makes the network more robust (Joo & Steinert-Threlkeld, 2018).

In conclusion, convolutional neural networks employ a series of convolutions and pooling operations to process images. This enables them to deal with the high-dimensionality, the redundancy and the spatial nature of image data. However, it would be suboptimal to use CNNs when processing tabular data. Since in the last model both images and tabular data are used, a network architecture which can deal with multiple data modalities is presented.

4.1.3 Introduction to multi-input neural networks

While deep feedforward neural networks and CNNs are widely used, multi-input neural networks are less famous. As suggested by the name, these architectures are able to take multiple inputs (Rosebrock, 2019). The inputs are fed into the networks thanks to multiple input branches. Every branch takes different data as input. The branches are composed by a series of layers which process the input. The input branches are joined together by a hidden layer, also called *concatenation layer*. The concatenation layer is followed by other hidden layers and by one output layer. Figure 2 gives a representation of the multi-input architecture employed in this thesis, it might help to understand the structure of these networks.

The strength of multi-input neural networks lies in the use of multiple input branches. In fact, the branches enable the networks to accept input data which differ in form, nature and modality. Moreover, every branch may employ different hidden layers. This way, the layer type which is more suited to process the input data can be used (Rosebrock, 2019). For instance, a branch processing images could employ convolutional layers, while another branch of the same network could use fully connected layers to process tabular data.

Multi-input neural networks, CNN and deep feedforward neural networks look very different. However, they are all trained in the same way, which is described in the next subsection.

4.1.4 Loss functions, backpropagation and optimization algorithms

All neural networks are composed by layers, all the layers are made up by nodes and all the nodes have some weights. To fit a neural network, it is necessary to find the optimal weights for all the nodes. Thus, the weights that minimise a loss function shall be found. The loss function $J(W)$ of an NN computed on n data points, can be written as:

$$J(W) = \frac{1}{n} \sum_{i=1}^{n} \lambda(f(x_i, W), y_i)$$

where W is the weight matrix of the neural network, x_i is the vector representing the ith data point, y_i is the real dependent variable for the ith sample, $f(x_i, W)$ is the formula notation for the NN and λ is the loss function (Friedman et al., 2001). λ needs to be chosen based on the problem at hand. Given that this thesis deals with a regression problem, the loss function

of choice will be the squared error. Thus, the loss function of the models used in this paper could be rewritten as:

$$J(W) = \frac{1}{n}\sum_{i=1}^{n}(f(x_i, W) - y_i)^2.$$

This is nothing but the mean squared difference between the value predicted by the NN and the real value of the dependent variable, also called *mean squared error* (MSE). Other loss functions can also be used for regression problems, including *mean absolute error* and *mean logarithmic squared error*. However, the MSE is commonly used for regression problems (Reed & Marks, 1999). Moreover, the mean squared error is the maximum likelihood estimator for the loss function when the target variable follows a Gaussian distribution, which is beneficial for machine learning models (Goodfellow et al., 2016). Hence, the MSE will be used as the loss function of choice.

To find the W which minimises the loss function, at first, the weights are randomly initialised, then an optimization algorithm iteratively updates the weights looking for the optimal weights' configuration. The backpropagation algorithm is at the base of this optimization process (Goodfellow et al., 2016). Indeed, backpropagation is used to compute the direction in which to update a weight. It does so by efficiently computing the gradient of the loss function with respect to the weights (i.e. $\frac{\delta J(W)}{\delta W}$). It is called backpropagation because it proceeds backwards. It starts by calculating the gradient of the output layer's weights and then it iterates backwards through the layers. Backpropagation uses the chain rule of calculus to compute the gradients (Goodfellow et al., 2016).

As said, backpropagation is used in gradient descent optimization algorithms, which iteratively modify the weight matrix according to the computed gradient. A dummy formula that helps to visualise how the weights are updated is given:

$$W^m \leftarrow W^{m-1} - \eta\frac{\delta J(W)}{\delta W}.$$

W^m indicates the new weight matrix after the mth iteration, while W^{m-1} is the old set of weights which is being updated. η, the so-called *learning rate* (LR) or step size, determines the magnitude of the update. There is a plethora of optimization algorithms. Some, such as *gradient descent*, *stochastic gradient descent*, *batch gradient descent*, use a fixed learning rate (Goodfellow et al., 2016). Others, such as *RMSprop* and *Adam* employ an adaptive LR, i.e. the learning rate changes throughout the learning process (Goodfellow et al., 2016).

Adam, in particular, is a very advanced optimization algorithm that uses several expedients to quickly find the optimal weights' configuration (Ruder, 2016). It is initialised with an initial learning rate. Then, it iteratively samples a subset of training data. This subsample is also called *batch*. For each sampled batch, it estimates the first and the second moment of the gradients through backpropagation and, based on these estimates, it performs an adaptive weight update (Ruder, 2016). To do so, it uses an adaptive learning rate approach which consist of computing new individual learning rates for each parameter at each step (Ruder, 2016). To be more effective, Adam takes into consideration the results of the previous weight

updates to compute new ones (Ruder, 2016). Thanks to this complex process, Adam avoids local minima and saddle points and converges fast without the need of extensive parameter tuning (Ruder, 2016). In fact, of Adam's several parameters, only the initial learning rate shall be tuned (Goodfellow et al., 2016). This is why Adam is considered one of the best choices for optimization algorithms (Ruder, 2016). Hence, in this thesis, the Adam optimization algorithm is used to fit all the models.

As mentioned before, Adam iteratively samples batches of training data without replacement. The number of data entries contained in a batch is determined by the *batch size*. After a number of iterations, the whole training set, divided into batches, is passed through the optimization algorithm exactly once. This is what is called an *epoch* in machine learning jargon. Optimization algorithms usually run until a predefined number of epochs is completed, i.e. until they have cycled through the entire training set for n times. However, to avoid overfitting, *early stopping* will be used in this thesis. Early stopping is a trivial regularization technique which relies on stopping the training process when no improvement on the validation loss is observed for a certain number of epochs, also called *patience*. This way, the neural networks do not learn a solution which fits well the training data but does not generalise well on the rest of the dataset, i.e. they avoid overfitting.

Weight decay is another regularization technique that will occasionally be used. Weight decay forces the weights of the network to shrink in proportion to their size at every weight update (Goodfellow, I. et al., 2016). Hence, it leads weights to be closer to 0. This, in turn, allows the model to generalise better (Goodfellow, I. et al., 2016).

Finally, it must be noticed that both the input and the target variable fed to the neural networks shall be scaled (Bishop, 1995). In fact, if the input data have different scales, the complexity of the problem increases, compromising the well-functioning of the fitting process. Moreover, if the input values are large, large weights must be learnt, which slows down the fitting procedure and may result in unstable models (Bishop, 1995). Secondly, large target values may cause large gradients which lead to very large weights updates, thus, hindering the learning process (Bishop, 1995). Therefore, in this thesis, the target values are standardised and the input values are standardised or normalised when needed.

Having gained a good understanding of neural networks, it can now be explained how these advanced machine learning techniques are used to tackle the problem at hand.

4.2 Unimodal models for numeric or categorical variables and for text data

Deep feedforward neural networks are used both for text data's and for numeric or categorical variables' unimodal regression. The unimodal model for categorical and numeric variables takes as input the standardised one-hot-encoded categorical features and the standardised numeric features. For what regards text data, instead, the *bag-of-words* assumption will be used. A *document-to-term matrix* (DTM) weighted by term frequency will be used to represent the text corpus. A document-to-term matrix is simply a matrix with as many columns as terms in a text corpus and as many rows as documents in the corpus. Calling this matrix M, the entry i, j of M will be the number of times the term j appears in the document i. Before creating the DTM, all the letters in the text are set to lower-case, punctuation and numbers are removed. Uninformative words, also called stop-words, are filtered out (e.g. articles and

pronouns). All remaining words are stemmed with the Porter algorithm. This means that the words are reduced to their root or stem so that terms with similar meanings are encoded in the same way. Then, less frequent words are pruned out and the standardised document-to-term matrix is fed into a neural network.

While there is no doubt that deep feedforward neural networks are among the best algorithms for performing regression tasks on numeric or categorical data (Friedman et al., 2001), many other approaches have been proposed for text data. However, bag-of-words and deep feedforward neural network have been proven to be among the most simple and yet effective ways to tackle such problems (Minaee et al., 2016). Furthermore, a simpler model is also easier to interpret. That is why other approaches, such as *word embeddings* and *recurrent neural networks*, will not be explored. This way, more resources can be focused on building the other models, especially the multimodal one. But, it is still worth to mention that the bag-of-words assumption is subjected to some limitations, for instance it does not consider the order or the context in which words are used in a sentence.

To conclude, deep feedforward neural networks are very powerful methods, but they are also complex to implement. Therefore, it is better to use them only when it is convenient. This is the reason why, for the first two unimodal models, a baseline linear model is created at first. The more advanced networks will be employed only if they will achieve better performances than the baseline models.

4.2.1 Linear baseline model

As a baseline model, in both the cases, a simple perceptron is used, i.e. a perceptron with a linear activation function which takes as input the input values and gives as output the price prediction. It is trained with Adam, with the default learning rate of 0.001, 1000 epochs and early stopping with a patience of 20. It is equivalent to a linear regression, but it is trained in a slightly different way. If this simple model was found to overperform the others, it would mean that the problem is better solved by a linear solution and that interactions between data are not relevant. Linear regression is not used as a baseline model because eventually all the unimodal models will be used to compose the multimodal model. Hence, it is necessary that the unimodal models are neural networks.

4.2.2 Deep feedforward neural networks

Deep feedforward neural networks shall perform better than the baseline model. However, it is more complex to configure them. In fact, many parameters determine the behaviour of these networks. In section 4.1.1, it was already outlined how the input and output layers shall be designed and what activation function shall be used in the hidden layers. In subsection 4.1.4, the rationale behind the choice of MSE as the loss function and Adam as the optimizer was given. In addition to this, the number of training epochs is set to 1000, an arbitrarily high number. On the other hand, the early stopping rule, with a patience of 20, is used to stop the training process. This patience will make sure that the maximum number of epochs is rarely reached. It will also speed up the hyperparameter tuning process and avoid overfitting, without hindering the fitting process. Indeed, if no improvement is observed for 20 epochs, it is most probable that no improvement will take place further on in the training process. The

same patience value is used throughout the thesis since the same reasoning holds for all the models.

Furthermore, *grid-search* is used to find the optimal parameters' configuration for the learning rate, decay, batch-size, number of hidden layers and number of nodes in the hidden layers. Hence, at first, the data at hand are divided into a training and a test set, with an 80/20 split. Then, 20% of the training data are used as the validation set. For every parameter, a series of possible values are drawn from a plausible parameter space. All the possible combinations of parameters are created. Models are iteratively fitted using different parameters' configurations. Eventually, the parameters' configuration which produces the lowest loss on the validation set is selected as optimal.

The values selected for the learning rate are 0.0001, 0.001 and 0.01. 0.001 is the default value proposed for Adam, the other two numbers are equally distant from it on a logarithmic scale (Goodfellow et al., 2016). This way, different values of the parameter LR are tested in order to find which one leads to the best performance. In particular, three values are compared: the default value of lr, a value which is significantly higher and one that is significantly lower. Moreover, it is usual to pick learning rate values on a logarithmic scale when doing grid-search (Goodfellow et al., 2016). For the decay, the values 0, 0.1, 0.01 and 0.001 are tested, where 0 means that there is no weight decay and 0.1 means that weights are iteratively reduced by 10% of their size. This way, models with different degrees of regularisation are tried. The proposed values are again equally spaced in a logarithmic scale. The batch-size shall be 32, 256 or 512, since it has been proven that values outside of this range are sub-optimal (Keskar et al., 2016). Both an architecture with one and another with two hidden layers are evaluated, given that more than one hidden layer is rarely necessary and two hidden layers can learn any function (Saeed & Snášel, 2014). A widely used rule of thumb says that the number of nodes of the hidden layers shall be between the size of the input layer (m) and the one of the output layer (Heaton, 2008). Hence, for each layer, the optimal number of nodes is searched among four values equally spaced in the interval [1, m].

4.2.3 Model-agnostic methods

To get a better understanding of which variables determine the optimal model's prediction, various model-agnostic techniques are used. First, the features' importance is assessed. Then, further analysis will uncover the effect of the important features on the model's prediction.

The *permutation feature importance* algorithm is used to measure the relevance of each variable. This algorithm is trivial but effective. It is based on one intuition: if a feature is important, the prediction accuracy is negatively affected when it is shuffled (Molnar, 2019). More specifically, to determine the feature importance, at first, the model's original prediction error on the test set is computed (e_{orig}). Then, iteratively, each variable is randomly permuted and the prediction error for the permuted dataset is computed ($e_{permuted}$). The feature importance of the jth feature is given by:

$$FI(X_j) = \frac{e_{permuted}}{e_{orig}}.$$

The higher the error of the permuted dataset, the higher is the feature importance. To increase the reliability of the feature importance, $FI(X_j)$ can be computed multiple times so that

confidence intervals can be built (Molnar, 2019). The test set is preferred to the train set in this analysis, because feature importance calculated on the train set may give more relevance to variables which are used to overfit data rather than to variables which really lead to good predictions.

Once the most important features have been found, their effect on the model's output shall be analysed. *Partial dependence plot* (PDP) and *individual conditional expectation* (ICE) are two techniques, based on similar premises, which can be useful for such a task (Molnar, 2019). They both take into consideration only one feature at a time. Then, for all the entries of the training dataset, they substitute the variable of choice with the lowest possible value for that feature. They feed the modified dataset to the model and store the resulting predictions (Molnar, 2019). The techniques proceed to substitute the feature column with the second lowest value in the range of possible values, they input the modified data in the model, they store the predictions, and so on and so forth (Molnar, 2019). At the end of this iterative process, a matrix M is produced with as many rows as the entries in the dataset and as many columns as possible values of the feature of choice. $M_{i,j}$ is the prediction output for the input vector corresponding to the ith entry in the training dataset, but with the variable of choice substituted with the jth values in the range of possible values for the considered feature. Partial dependence plot and individual conditional expectation differ only in how they display the results for such a matrix. The partial dependence plot displays the mean value for every column, ICE plots all the rows separately. In the case of numeric variables, the results are plotted with line plots. ICE displays one line for each entry in the training set, while PDP displays only one single line. In the case of binary or categorical variables, boxplots are more suitable. These boxplots can be considered something in the middle between PDP and ICE. Indeed, they do not show the evolution of singular instance as ICE does, but they also do not show only the mean of the predictions as PDP does. To conclude, the rationale behind the two methods is similar, they show how the change in the value of the feature influences the final output. Moreover, they can also describe nonlinear relationships (Molnar, 2019). PDP has the advantage of giving more understandable representations, while ICE can be more insightful (Molnar, 2019). Both these methods do not show the effect of possible interactions between data.

Hence, permutation feature importance together with partial dependence plot and individual conditional expectation are helpful in interpreting the two unimodal models. This provides more insights to answer to the research question.

4.3 Image data unimodal model

The third model is the unimodal model for image data. As discussed in section 4.1.2, convolutional neural networks are better suited to analyse image data. CNNs are the state-of-the-art technique in the research field of computer vision (Joo & Steinert-Threlkeld, 2018). Therefore, this kind of networks are used to create a unimodal model that is able to predict the price of a horse from its image. However, CNNs' architectures are far more complex than the architectures of feedforward neural networks. The most efficient CNNs have several layers and millions of parameters (Alom et al., 2018). Moreover, over the years, the most diverse normalization techniques and layer organizations for CNNs have been invented (Alom et al., 2018). Thus, architecture search for CNNs is extremely complicated and requires high

computational power: there are infinite combinations of layers and hyperparameters to be tested. Moreover, given the high number of parameters, millions of images are needed to fit a CNN model without incurring in overfitting.

From these premises, it would be impossible to find a CNN architecture that optimally solves the problem at hand and to train it, both because the training set is limited (~5000 entries) and because the hardware at our disposal is not the most advanced. However, it has been proven that CNN architectures which perform well on the Imagenet task, also perform well when applied to other problems (Kornblith et al., 2019). Moreover, when there are limited data to fit the algorithm, using the weights fitted on Imagenet as a starting point and slightly modifying or *fine-tuning* them to tackle another task, is proven to be an effective strategy (Kornblith et al., 2019). The procedure of using a CNN that was originally trained for another problem to solve a new task is called *transfer learning*.

Transfer learning works because the first convolutional layers of CNNs learn filters which extract high level features that are not task-specific and can be useful to solve many different problems (Yosinski et al., 2014). Deeper layers are more task-specific, instead (Yosinski et al., 2014). Transfer learning of CNN has already been successfully employed in a price prediction problem (Chen et al., 2018).

There are many approaches to transfer learning (Yosinski et al., 2014). In this thesis, a CNN trained on Imagenet is used as a base model. Since the Imagenet task is a classification task, the head of the model is substituted so that it can perform a regression. Even if for both the tasks the input are images and, often, pictures of animals, the tasks are still quite different. Hence, an attempt is made to fine-tune some of the low-level filters. Not all the filters are fine-tuned, both because the high-level filters shall work well for any task and because the training dataset is limited. Hence, fine-tuning many parameters on it would lead to overfitting. Moreover, the computational power at hand is limited, thus, it is better to reduce the number of parameters to train.

For succesfully doing transfer learning, it is crucial to select a good starting model to further fine-tune. In this case, the optimal model should have two characteristics. First, it shall perform well on Imagenet, since, as previously explained, this is an indicator that the CNN can perform well on other problems (Kornblith et al., 2019). Second, it shall be efficient in terms of computational time, given the limited hardware resources at disposal. *EfficientNet-B0* perfectly meets these two criteria. In fact, it belongs to EfficientNet, a family of convolutional networks which achieves state-of-the-art accuracy on Imagenet, while having a limited number of parameters and being fast in terms of floating point operations per second (Tan & Le, 2019). EfficientNets have also been proven to perform well when used for transfer learning (Tan & Le, 2019). EfficientNet-B0 is by far the most efficient model in the family, but it still performs well in terms of accuracy. Thus, EfficientNet-B0 with weights pre-trained on Imagenet is the base model of choice for this thesis. Moreover, in some early experiments, it was discovered that switching to bigger architectures of the same family did not improve the end results. Early experiments with other widely used CNNs were done as well, but they lead to significantly worse performances.

4.3.1 Model fitting and hyperparameters tuning

The hyperparameters of EfficientNet-B0 shall be optimized. For the previous two unimodal models, the optimal hyperparameters' configuration was found by the extensive use of grid-search. Such a procedure would not be feasible for this model. Indeed, the computational complexity of the CNN slows down the fitting procedure, thus it is not possible to test 100 different combinations of parameters. Therefore, the model training and tuning procedure described in this subsection will carefully balance the importance of finding the optimal parameter configuration and the necessity of not overcoming the hardware limits.

Luckily, the optimal architecture for the model is already known. Only the output layer of EfficientNet-B0 is modified. The output layer is substituted by a fully connected layer with a single node and a linear activation function. No additional architecture search is done. The input size is easy to determine as well. Indeed, the optimal input size for EfficientNet-B0 are images with a resolution of 224×224 encoded in an RGB colour space, with the colour values normalised to be in an interval [0, 1]. Moreover, it was tested that a higher resolution does not improve the model's performances. To avoid overfitting, the images of the training set are randomly transformed. This technique is also called *image augmentation*.

Since all the layers of the neural network beside the last one have pre-trained weights, at first, only the weights of the last layer are trained. Thus, all the other weights are frozen. This first fitting step produces a baseline model, where a linear regression is applied to features extracted from the images. In a second step, this baseline model will be further fine-tuned, on the look for a better weights' configuration. So, since the problem to fit in this first step is not very complex, the value of the learning rate, batch-size, decay and epochs are not tuned. The default LR for Adam (0.001) is used. A batch-size of 32 is chosen so that the RAM is not overflooded. This model has few trainable parameters, so it is unlikely to overfit. Thus, no regularisation is needed and the decay is set to 0. The maximum number of epochs is 200 and early stopping with a patience of 20 is used to cut computational time. The running time per epoch is higher than the one of the previous models, so it would be unfeasible to let a model run for the same number of epochs as before (i.e. 1000). The best model in terms of validation loss is saved at each epoch and is used for the next steps. This is a precaution against overfitting and hardware malfunctions.

For the final fine-tuning step, different parameters' configurations are tested. The one achieving the best results on the validation set is used in the final model. The number of epochs is 100 and early stopping with a patience of 20 is used again. The maximum number of epochs is further reduced, since the running time per epoch increases. The batch-size remains 32, since bigger bacthes would overflood the RAM. Moreover, in early tries, it has been seen that slightly bigger batches do not improve the end results. Grid-search is used to test different combinations of learning rates and numbers of unfrozen layers. Since the operation of fine-tuning aims to slightly modifying the pre-trained weights, very small LRs and no decay are used. Learning rates that are too big would lead to big updates which would modify the pre-trained weights drastically, thus losing the advantages of transfer learning. Similarly, the weight decay would iteratively shrink the model's weights, with the same consequences. The tested learning rates' values are 0.0001, 0.00001 and 0.000001. The highest of the tested values is one order of magnitude smaller than the default value for Adam, 0.001, hence it is

significantly low. The other tested values are even lower. All the values are equally spaced in a logarithmic scale. Considering that EfficientNet-B0 has got seven convolutional blocks followed by an output layer, three different ways to unfreeze the layers are tried: unfreezing only the last convolutional block, unfreezing the last three convolutional blocks and unfreezing every layer after the second convolutional block. Unfreezing more blocks is not possible due to hardware limits. Still, unfreezing all the blocks after the second one allows the retraining of 98% of the model's weights.

4.3.2 Model-agnostic methods for convolutional neural networks

It would not make sense to apply permutation feature importance, PDP or ICE to image data. Indeed, a change in the value of a single entry in the three-dimensional matrix would not influence the overall prediction. Moreover, the pixels cannot be compared based on their position in the matrix. Again, image data are peculiar and they require data-specific model-agnostic techniques.

Among the many model-agnostic techniques for CNNs proposed in literature, only the one of sliding window heatmaps can be applied to regression problems (Chen et al., 2018; Ribeiro et al., 2016). Sliding window heatmaps, also called *occlusion maps*, give local model-explanations, which means that they give a figurative interpretation of which part of a single image determines the model's outcome. They do so thanks to a simple algorithm. For a selected image, at first, the price-prediction of the model for that image is computed ($\hat{y}_{original}$). Then, the picture is divided into a grid of squares. Iteratively, every square is greyed out and the prediction $\hat{y}_{occluded}$ for the picture with the occluded square is observed (Chen et al., 2018). The effect (e) of the occluded square over the model output is computed as:

$$e = \hat{y}_{occluded} - \hat{y}_{original}.$$

e is positive when the occlusion of the part of the picture raises the predicted value and vice versa. In this thesis, the pictures have a resolution of 224×224 and they will be divided in squares of 28×28 pixels, in total 64 different windows are considered. The pixels' value of the occluded window will be set to the average pixel value of the pictures in the training set. For every picture, 64 different e's are given, which are collected in a vector E. The most effective way to visualise e is with a colour-coded heatmap. A blue-grey-red colour scale is used. For instance, parts of the picture which, when obscured, raise the prediction value (i.e. with high e), will be coloured in red. It can be said that the parts in red negatively affect the predicted price, the ones in blue have a positive effect instead.

This technique can lead to uncovering very interesting insights. However, it is bound to give local interpretations. It does not give a global understanding of the model; it only explains how certain parts of a single image can explain a single prediction. In the next subsection, it is discussed how to use local explanations to build a global understanding of the model.

4.3.2.1 From local explanations to global understanding

Intuitively, by observing many local explanations, a global understanding of the model could be reached (Ribeiro et al., 2016). However, since human time, patience and attention is

limited, it is feasible to consider only a few local explanations which need to be picked wisely. Ribeiro et al. (2016) provide a more formal outline of the problem and suggest some guidelines to solve it. They advise giving explanation for a number B of pictures which is enough to make the model trustworthy to the reader without consuming too much of its patience (Ribeiro et al., 2016). In this thesis, B is set to 20. In fact, it is argued that it would be hard for the reader to look at, temporarily memorise and successfully analyse more than 20 different pictures. Moreover, Ribeiro et al. (2016) suggest using a selection rule to pick explanations which are insightful and not redundant. They provide an example of pick-up rule for tabular data which cannot be applied to image-data. In this thesis, instead, at first, explanations are produced for every element in the test set (~1000). Then, the variance of the absolute values of the explanation vectors (E) is computed. The twenty explanations with the highest $var(|E|)$ are picked. This way, pictures in which a relevant explanation is given by around half of the windows are selected. Only insightful and clear explanations are shown.

4.4 Multimodal fusion model

The fourth model, as already mentioned, is a multimodal fusion model which consists of a multi-input neural network. The model has three input branches joint together in a hidden layer to eventually produce a single output. The three input branches resemble the unimodal models described in the previous sections in their architecture and function, since one branch accepts as input the numerical and categorical variables, another is fed with text data and the last one processes images. The first two branches are formed by fully connected layers, while the branch processing images makes use of convolutional layers.

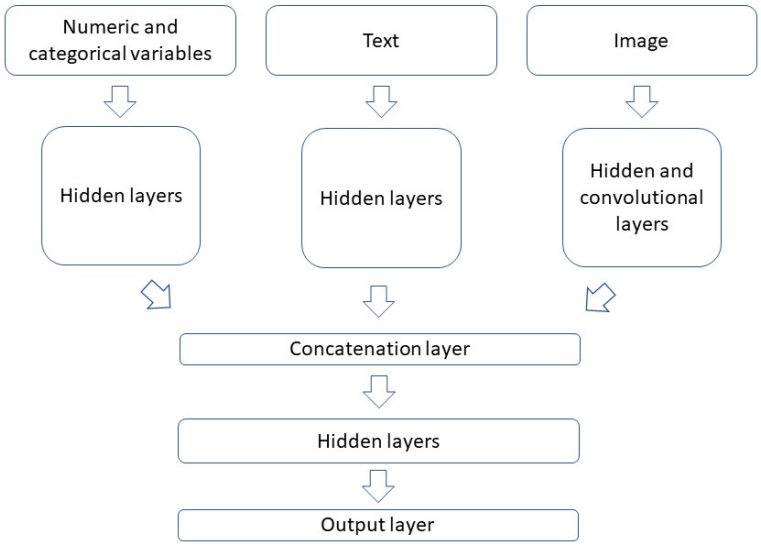

Figure 2: Network architecture for the multimodal fusion model.

The three branches are concatenated in a shared joint hidden layer. The concatenation layer accepts as input the output of the three input branches, and simply concatenates it in a unique tensor. In this layer, features extracted from data of different modalities by the input branches are joint first. The concatenation layer is followed by other hidden layers, which further process the input, allowing interactions between the data of the different modalities. Finally, an output layer with a single node and a linear activation function ends the model. These last layers can be called fusion or decision layers. In fact, in these layers, different data-modalities are fused together to take an end-decision.

Figure 2 helps to visualise the proposed architecture. Such an architecture is advantageous under many points of view. First, the three different branches allow one to separately process inputs which are inherently different and to extract compatible feature representations from the data. Moreover, using three different pre-processing branches enables using the network architecture which better suits a specific type of data, i.e. convolutional layers. Additionally, the final part of the model combines the input coming from all the available data sources to make a prediction. Finally, the fact that the three unimodal input branches and the multimodal regression layers are combined in a single network, allows for a conjoint training of them. In other words, during the training process, the three input branches learn unimodal data representations which are best suited to be an input of the multimodal decision layers. Simultaneously, the fusion layers learn the weights which can better exploit the unimodal data representations to predict accurately. The only disadvantage is that it is hard to interpret this model. Indeed, the model-agnostic techniques used for text and categorical or numeric variables cannot be used for image data and viceversa. Furthermore, little research has been done on how to interpret multimodal neural networks (Baltrušaitis et al., 2018).

The aforementioned advantages shall highlight once more why neural networks were chosen over tree-based methods. Tree-based methods would not have been able to process directly all the multimodal data. Some generic feature extractors would be needed to reduce the dimensions of the images and the text data. As previously explained, such an operation is not necessary with multi-input neural networks. Moreover, tree-based methods are not easier to interpret than NNs. Still, developing a multi-input neural network model can be quite challenging.

4.4.1 Architecture search and hyperparameter tuning

The multimodal fusion model is more complex than any other model described so far. The architectures that could be used for it are practically infinite. At the same time, fitting this model is computationally more complex than training a CNN. Thus, it is paramount that the process for finding the best model architecture and the best set of hyperparameters could also run in a reasonable amount of time.

To start, the architectures of the three input branches is set to be equal to the ones found to be optimal for the three unimodal models, without the output layer. For example, if the optimal architecture for the text data is found to have one hidden layer with 100 nodes, then the input branch for text data will be composed by an input layer followed by a hidden layer with 100 nodes, whose output will be input into the concatenation layer. This is done based on the rationale that if the architectures were found optimal for the unimodal models, it

means that they were able to extract useful features for the regression tasks. Moreover, this procedure allows one to avoid a long architecture search process to find the best input-branches configuration. Finally, this implies that the fitting process is heavily influenced by the nature of the image branch. If it is found that it is better to fine-tune the weight of the CNN for image data, than the image branch will use the EfficientNet-B0 architecture with pre-trained weights. Instead, in case fine-tuning was found not to improve the CNN's performance, the image branch will consist of a single input layer taking the image features extracted by the pre-trained convolutional blocks as input. Two different approaches are discussed for the two eventualities.

In the first case, since fine-tuning would be necessary, the fitting process for the multimodal fusion model would consist of two steps, a first one in which only the non-pre-trained weights are trained, and a second one in which the pre-trained weights are fine-tuned. This complicates the hyperparameter tuning process. To simplify the problem, the batch-size is set to 32, which is a number low enough not to overflood the RAM. Furthermore, the number of epochs used is 200 together with the early stopping rule with a patience of 20. Since the running time per iteration is high, it would be unfeasible to let any model run for more than 200 epochs. Keeping the CNN weights frozen, grid-search is used to find the best architecture for the decision layers. Three different architectures will be tried, one with no hidden layers and two with one hidden layer. The latter two will have a number of nodes equal to respectively 1/3 and 2/3 of the nodes of the concatenation layer. So, the number of nodes will remain between the number of the layer's input and output values, as suggested by the rule of thumb (see section 4.2.2). In this grid-search, a starting LR of 0.001 and 0.0001 will be tried. The first number is the default value for Adam, the latter is one order of magnitude smaller, testing a third value would be too computationally expensive. Three values of decay will be tested: 0, 0.001 and 0.1. These values correspond to different degrees of regularisation. As a final step, some layers of the CNN will be unfrozen and the overall model will be fine-tuned. The unfrozen layers are the ones that were found to be optimal to unfreeze during the training of the unimodal model for image data. It is indeed argued that, if it was found convenient to fine-tune the filters in those layers for the unimodal model, also the multimodal model could benefit from it. Two different learning rates are tried in the fine-tuning: 0.00001 and 0.000001. As mentioned in section 4.3.1, when fine-tuning, the learning rates need to be low, i.e. some order of magnitudes lower than the default value (0.001). No decay is used, in order not to drastically modify the pre-trained weights. At the end of this complex optimization procedure, the model is finally fit.

In the second case, in which EfficientNet-B0 would be used as a mere feature extractor, the complexity of the fitting process would decrease greatly. In fact, before fitting the model, the images would be reduced to a vector of features through the pre-trained convolutional blocks. Therefore, big batches of images would not overflood the RAM. Moreover, there would not be any need to fine-tune several millions of parameters. Thus, the whole model could be fitted into a single step and the computational complexity of the problem would drastically decrease. Hence, in this second scenario, an extensive use of grid-search can be employed to find the best parameters' configuration for the learning rate, decay, batch-size, number of fusion layers and number of nodes in the fusion layers. The tested learning rates are 0.0001, 0.001 and 0.01. Four different values are tried for decay: 0, 0.1, 0.01 and 0.001. The optimal batch-size

is searched among 32, 256 and 512. The possible values for these parameters are chosen following the same rationale as in section 4.2.2. Architectures are tested with both zero and one hidden layers between the concatenation layer and the output head. Experimenting with more than one hidden fusion layer made the computer crash. For the number of nodes of the hidden layer, four different values are tried which are equally distributed between one and the number of nodes in the concatenation layer. For the grid-search, the number of epochs is fixed to 1000 and the patience of the early stopping rule to 20. Again, these two parameters are equal to the ones used in section 4.2.2.

As a final remark, it could be argued that a single input branch could be used both for text data and numeric or categorical variables. However, even if similar architectures are used for both the data modalities, the data per se are quite different, perhaps the DTM is very sparse. Hence, it is better to use two different input branches, which can use different numbers of nodes and hidden layers. This completes the detailed explanation of the four proposed models.

4.5 Model evaluation

It is important to compare the performance of the four models described above. To do so, the same ads are used to create the training, validation and test set for all four models. This implies that only the ads with an image are considered for all four models. The accuracy of the models is measured in terms of mean squared error, which is also the loss function used to tune all the models. The MSE is computed on the test set so that it is not influenced by overfitting. The formula of the MSE is:

$$\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2$$

where y_i is the value of the independent variable for sample i, and $\hat{y}_i$ indicates the estimate of y_i made by the model. Additionally, to test if the mean squared errors of two different models are significantly different, t-tests are used. Instead, one-way analysis of variance is used to test if the MSEs of multiple models are statistically different. Moreover, since the independent variable is first transformed using the logarithmic function and then scaled, a real-valued *root-mean-square error* (RMSE) is provided as well by applying reverse transformations to the model's predictions. The RMSE is nothing but the square root of the MSE. Furthermore, partial residual plots are used to investigate the relationship between the value of the response variable and the squared residuals. The coefficient of determination (R^2) is also computed on the test set for all the models. This adds extra insights to interpret the precision of the designed models. The formula for R^2 is:

$$1 - \frac{\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$$

where y_i and $\hat{y}_i$ are the same as before, and $\bar{y}$ is the mean of y. In short, the R^2 gives the proportion of the variance of the response which is explained by the model. The closer to one, the better it is.

5 Results

5.1 Unimodal model for numeric or categorical variables

All the numeric or categorical variables at disposal are input into the first unimodal model. Five numeric variables are used: *Height*, *Weight*, *FoalDate*, *Temperament* and *noOfPictures*. Five binary variables are taken from the database as well: *InFoal*, *hasVideo*, *hasPedigree*, *hasShippingNotes* and *hasOwnerDescription*. Eight categorical variables are one-hot encoded: *currency*, *typeOfAd*, *StateBred*, *Gender*, *location*, *Breed*, *Registry* and *Color*. The less frequent levels of these variables are set to be "Others", so that features with too little data to be informative are not included in the input. In total, the number of levels for *StateBred*, *location*, *Breed* and *Registry* is reduced to 50. Moreover, only the 34 most frequent colours are considered. All the levels of the remaining three categorical variables are used as input. Finally, additional variables are obtained from *Markings* and *discipline*. For what regards *Markings*, a dummy variable which indicates if a horse has one of the most common horse markings is created. The markings accounted for are: *sock*, *star*, *snip*, *blaze*, *stock*, *tobiano* and *None*, if none is given. The colour of the marking is not considered and a horse can also have multiple markings. Similarly, for *discipline*, dummies are created for the 30 most common skills and disciplines including *jumping*, *dressage* and *trail riding*.

Eventually, the predictors matrix counts 293 columns. For all the categorical variables, a base category is kept to avoid the presence of perfect collinearity in the data. The labels consist of one column: the logarithmic transformation of the horses' prices (*logPrices*). Given that only the entries with a picture are considered, the total number of samples is 6543. The data is split into training, validation and test sets, with respectively 4274, 997 and 1272 entries each. Both the labels and the predictors matrix are standardised. The mean and the standard deviation to standardise the data are only computed over the training and validation set, and then used on the test. This way, the test data are never used in the training process and can thus be used to effectively test the performance of the model over new data.

The linear baseline model reaches an MSE of 0.56 over the validation set. Grid-search is used to see if deeper neural networks would perform better. The values tested for the number of nodes in the hidden layers are 37, 110, 183 and 256. These values are equally spaced in the interval delimited by the output size (1) and the input size (293). In total, 720 different parameters' configurations are tested through grid-search; 144 neural networks with only one hidden layer and 576 with two hidden layers. It is found that the neural networks with a single hidden layer do not outperform the baseline model, achieving in the best case an MSE of 0.56 on the validation set. The best parameter configuration for neural networks with two hidden layers reaches the best performances so far with an MSE of 0.52 on the validation set. Overall, the comparison between the baseline model and the neural networks shows that a simple linear model can, sometimes, overperform other more complex models. This suggest that many of the predictors are linearly related to the logarithm of the price. Hence, in the case in which a simpler and more interpretable solution was needed to solve this problem, a linear regression could be used. However, in this thesis, the model that performs the best is preferred, even if more complex.

Thus, the optimal model is found to be a neural network with two hidden layers, the first with 256 nodes and the second with 183 nodes. This model is trained with a learning rate

of 0.01, a decay of 0.1 and a batch-size of 512. In fact, these are the hyperparameters that grid-search finds to minimise the loss on the validation set. The decay value is high, meaning that the network is strongly regularised. Still, the model slightly overfits having a loss of 0.33 on the training set and 0.52 on the validation set.

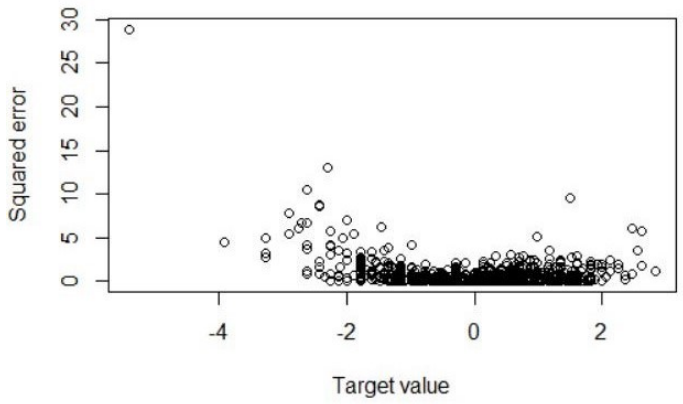

Figure 3: Scatterplot of squared residuals over the logarithmic standardised price.

The MSE of the optimal model over the test set is 0.548, the R^2 is 0.415. The R^2 is in line with the literature, as Freeborn (2009) had one of 0.33. Freeborn (2009) uses linear regression for a hedonic pricing analysis on similar data, therefore it is ideal to compare this model with hers. Squared residuals are distributed almost uniformely with the exception of some outliers at the two ends of the distribution (see Figure 3). The RMSE in dollar terms is $5397.

Explaining the functioning of the optimal model is paramount to answer the research question. Table 2 shows the results of the feature importance permutation. Every variable is permuted 50 times to obtain reliable results. 131 features are found to be important in determining the results. 57 features improve the loss of more than 1%. 22 features improve it of more than 2%. For the sake of clarity, only the 125 most important variables are reported in the table. Among the most important features are some numeric variables describing the phenotypic characteristics of the horses, including, in order of importance, *Weight*, *FoalDate*, *Height* and *Temperament*. Moreover, there are some variables describing the type of ad posted by the seller: *hasVideo* and *typeOfAdstandard.photo.ad*. Many dummies related to the breed result to be important: there are four breed categories among the ten most important variables. Another type of genotypic information which is relevant in predicting the price is the gender of the horse. Finally, some dummies for *Registry*, *discipline*, *StateBred*, *location* and *Markings* are relevant, even if to a somewhat lower extent.

The variables mentioned in the previous paragraph shall be considered as the most important horses' price determinants. To better understand their effect on the price prediction, partial dependence plots and individual conditional expectations are used. To start with, the effect of numeric variables over predictions is inspected. Figure 4, 5, 6 and 7 show the partial dependence plots for, respectively, *Weight*, *FoalDate*, *Height* and *Temperament*. All these plots show very linear monotonic relationships between the numeric variables and the average value predicted by the model. This is in line with the fact that the baseline model could reach

good performances, even if it was a linear model. More in particular, the plots show that buyers are willing to pay more for bigger horses, both in terms of height and weight. Old horses are less valuable, probably because their physical fitness deperishes with age. Finally, people seem to prefer docile animals to more firey ones, which is in line with previous research on the value of safety in the horse market (Oddie et al., 2014). PDPs are used for the sake of clarity, the black ticks at the bottom of the graphs indicate where the original values of the data layed.

Genotypic features, as well as the previously discussed phenotypic ones, are important price determinants. Certain breeds seem to be more valuable than others. In fact, Figure 8 shows the effect of a sample of the 50 different breeds on the predicted value. All of the most important breeds, according to the permutation test, are included. *Others* is the baseline category for breed. The breed gypsy vanner and andalusian lead to higher prices. Donkeys and miniature horses, on the other hand, are considered less valuable. A similar boxplot is made for *Gender* variables (see Figure 9). It shows that *gelding* is not only one of the most important features, but also the gender that is related to the highest price predictions. This is in line with previous findings (Oddie et al., 2014). It is also worth noticing that gender categories indicating untamed horses are generally associated with lower prices (see *yearling*, *filly*, *foal* and *unborn foal*).

Figure 10 and 11 describe the effect of *typeOfAd* and *hasVideo*. Generally speaking, standard ads seem to be related to slightly lower price predictions. Having a video has a positive effect on the prediction. In addition to this, it is found that if a horse is considered safe for children, if it has got a blaze marking or if it is not fit for English pleasure riding, its price increases (see Figures 12 to 14).

The other features which are found to be important by the permutation test, do not restitute interpretable results when analysed with ICE and PDP. Hence, their plot is not included. This does not mean that they are not important, it only means that they are not connected to the dependent variable by an explainable relationship. Perhaps, their effect on the end variable depends strictly on the interaction with other variables, hence it can't be described by a boxplot.

To conclude, there are several findings in this subsection. First, the numeric and categorical information contained in the information section of the online ad is relevant in determining the price. Thus, it is worth listing this information thoroughly and correctly when selling a horse. Second, some of the features are more important price determinants than others. Third, among the relevant price determinants, there are the breed and the gender of the horse. This insight can be useful for horse farmers who want to develop the most valuable products. For instance, they could focus on breeding only the most valuable breeds of horses or they could castrate male horses, given that geldings are more valuable than stallions. Moreover, it is useful for horse traders who face a pricing decision. Some examples of how phenotypic, genotypic and characterial characteristics influence the price of a horse were given. Finally, these findings could be used to better promote the horses' sale online. In fact, it is demonstrated that having a video and paying for a non-standard ad has a positive effect on the specimens' prices. Still, there is space to improve the model's precision, and this will be attempted in the next subsections.

5.2 Text data unimodal model

The second model considers the text contained in the ads. Four different columns of the database contain text data: *smallDescription*, *AdditionalComments*, *ShippingNotes* and *ownerDescription*. The text in these four columns is pasted together to form one single text description for every ad. The text is then set to lower-case, the numbers and the punctuation in it are removed, English stop-words are deleted and all the remaining words are stemmed with Porter algorithm, as explained in subsection 4.2. The number of words per ad in the cleansed text corpus is computed and temprorarily saved in a vector called *numberOfWords*. On average, 80 words are present in each ad, having excluded stop-words. 18178 different terms are contained in the cleansed corpus. Only 1041 terms appear in at least 65 different documents, i.e. 1% of the total amount of ads. The most frequent term is *hors*, it is used 14946 times. Both the less frequent terms and *hors* are considered uninformative and pruned out. A document-to-term matrix is eventually created considering the remaining 1040 terms. These first 1040 features, together with *numberOfWords*, form the input to the text data unimodal model. It could be argued that, given the high number of features, it would be better to reduce the input dimensionality by principal components analysis or similar techniques. However, this would make it more difficult to uncover how a word relates to the model's output. Moreover, deep neural networks are able to deal with highly dimensional data (Goodfellow et al., 2016).

The input data are split again into a training, a validation and a test set. The ads are split in the same exact way in which they were split for the previous model, meaning that if the categorical and numeric variables of an ad previously belonged to the training set, the text data of that ad are now used in the training set. The input matrix and the independent variable (*logPrice*) are standardised.

The linear baseline model's loss on the validation set is 0.7. In the attempt to find a model which outperforms the baseline one, 720 different parameters' configurations are tested in the training process. In this case, the number of nodes in the hidden layers may take up these values: 130, 390, 650 and 910. Again, these numbers are equally spaced in the interval [1, 1041]. Tested architectures with only one hidden layer outperform both the baseline model and the first unimodal model with an MSE of 0.51. However, the best result is reached by a model with two hidden layers. The first hidden layer has 650 nodes and the second 390 nodes. It is trained with a learning rate of 0.01, a batch-size of 32 and no decay. Its mean squared error on the validation set is 0.44, the best one so far. The model performs well on the test set with a mean squared error of 0.45. Some more regularised architectures, i.e. with a higher decay, were tested during the hyperparameter tuning process, but they were found to underperform compared to the optimal one.

The text data unimodal model improves over the MSE of the first model of almost 20%, the two mean squared errors are proven to be different with a confidence interval of 0.9 by a t-test. The squared residual distribution is similar to the one of the previous model (see Figure 15). The RMSE in dollar value is \$5229. The R^2 is 0.51. All these metrics show some improvement over the previous model and over the previous literature.

To answer to the research question, however, it is important to analyse how this well-performing model makes predictions. Thus, the permutation feature importance is run. Since there are more variables than in the previous model, this time, only 10 different permutations are made

for each feature. There are 459 variables whose permutation has a negative effect on the loss function on average, 332 cause an average percentage change of the loss function of more than 1%, 224 of more than 2%. Table 3 shows the results of the feature importance permutation test for the top 224 features. The other features are omitted for the sake of clarity.

numberOfWords, the only continuous feature, is among the most important features. It is quickly assessed through PDP that there is a positive relationship between the number of words used in the ad and the price predicted by the model (see Figure 16). The other most important features are displayed in Table 4, together with their importance and their effect. The effects of the features are computed by taking the difference between the average value of the predictions when a term is set to the maximum observed frequency for every entry and when it is set to the lowest, i.e. when the word is never used in the ad. This mechanism is similar to partial dependence analysis but it ignores intermediate values, assuming the modelled relationship is linear. This assumption allow us to summarise many results better.

Table 4 is very insightful. It confirms many of the results obtained by the previous model. First of all, there are many terms related to the breed of the horse, including "miniatur", "poni", "andalusian", "friesian", "warmblood", "spanish", "foundat", "appaloosa" and "grade" (which are the stemmed forms of the original words). Once again, andalusian and friesian horses prompt higher prices, while ponies and miniature horses are less expensive. Moreover, the term "grade", which indicates horses with unknown parentage, is associated with cheaper animals. Secondly, terms related to the skills and disciplines of the specimen are deemed important as well. Being suitabe for western riding, able to jump, to work with cattle, to do track racing, having a good stride, having been tamed or trained are all qualities which raise the price of the specimen (see respectively "west", "jump", "cattl", "track", "stride", "break" and "train"). All around horses, instead, are generally cheaper (see "around"). The fact that referring to a video in the ad description prompts a higher predicted value is in line with previous findings (see "video"). Belonging to a register is also an important factor for this model. In particular, "amhr" and "amha", registries for miniature ponies, are associated to lower prices. Finally, the gender of the horse is found important again (see "mare").

Beside confirming previous findings, Table 4 gives some new insights. In fact, it shows that many terms related to the trading agreements are relevant for the model's accuracy. For instance, "leas" and "free" are negatively correlated with the price prediction, indicating that it is cheaper to get a horse on lease or, obviously, for free. Moreover, when "board" and "haul" options are discussed in the ads, the price raises. When the seller mentions in the ad that the horse has been "sold", the prices are usually lower, which can be explained by the fact that lower prices are cleared by the market first. Also the verb "pay" is important and it is positively correlated with the price. Another topic to which important terms often belong, is the one regarding the animals' health. Animals are preferred to be healthy and without any injury (see "healthi", "injuri"). Worm problems, when mentioned, have a relevant and often negative impact on the output (see "worm" and "deworm"). Quite surprisingly, different shades of coat colours are considered relevant. In particular, "tobiano", "buckskin" and "perlino" coats are positively valued, while "dark" coats are not. Finally, many adjectives are present in the table with alternating effects, including some which are referred to the horse as "pretti" (i.e. pretty) and some to the level of experience of the rider as "advanc", i.e. advanced.

Some of the relevant terms do not seem to have a strong effect on the outcome. However, this does not mean that their effect is insignificant, but that they influence the outcome in a way which cannot be captured by the partial dependence analysis, perhaps due to the presence of interactions. An exemplar case is the one of "fanci" (i.e. fancy), the most important term. Figure 17 shows that the more times "fanci" is present, the lower the variance of the predicted values is, while the mean remains unchaged. The effect is not positive or negative, but the term remains important.

To conclude, the unimodal model for text data shows interesting results. First, it is found that text data can be used to predict the price with a precision which is even higher than the previous model's one. Thus, it is important for the ad poster to write a text description that contains good information and a proper level of detail in order to show the real value of the horse to the buyer. Moreover, it means that horse-traders shall rely on the text description to understand the real value of the animal, i.e. for pricing decisions. Second, it is shown that longer descriptions are needed for more expensive specimen. Thus, if a seller wants to correctly promote an expensive horse, he or she shall give a detailed description of the animal which is able to justify why a higher price is asked. Moreover, it is proven again that the use of a video is a good promotion practice to advertise expensive animals. Breeds, coat colours, skills, disciplines and genders of the horses are found to be highly correlated with the end price. This can be useful to horse breeders who want to breed and train the most profitable animals, i.e. develop a good product. Furthermore, it seems important to discuss the horse's health status and the contractual arrangements for the purchase in the ad. Hence, a correct promotional text shall not omit these details. Finally, it is found that it is possible to detect which words are deemed relevant price determinants and to uncover their effect on the prediction output. Based on this result, the website could develop a tool that suggests to sellers which words to include in the description of their horse, in order to have a more effective ad.

5.3 Image data unimodal model

The third and last unimodal model is the image data one. 6543 ads displayed at least one picture. These ads are splitted into training, validation and test sets, like for the previous two models. As already mentioned, each picture is resized to be 224×224 pixels. Every picture is then encoded into a three-dimensional matrix. Every entry in the matrix is a float in the interval [0, 1].

The training process for this unimodal model is more complex than that of the previous models. It is organised in two steps. In the first step, only the head of the model is trained, i.e. only the weights of the output layer are fitted. The obtained model is the baseline model. The optimal weights' configuration of this baseline model has an MSE of 0.77 over the validation set. Nine different attempts to further improve the model by fine-tuning are done. None of them leads to any further improvement in the validation loss.

There could be many reasons why the baseline model is able to out-perform the fine-tuned ones. The pre-trained weights might already be optimal for the problem at hand. There could also be too little data to properly fine-tune millions of parameters. Or, the picture of a horse might not contain enough information to further improve the price prediction. Anyway, the

baseline model is certainly the best performing one, hence its results are analysed further.

The baseline model's MSE on the test set is 0.78, which is significantly higher than the one of the first unimodal model according to a t-test. Its real-valued root-mean-square error is $6690. Figure 18 shows the distribution of the residuals. The errors grow as the price moves away from the mean. This might suggest that the model tends to predict prices close to the mean price to minimise the loss. However, the R^2 is 0.16, indicating that the predictions differ from the simple mean value to a certain extent.

The model-agnostic techniques can give some insights into how the model produces the predictions. Figures 19 to 38 show the twenty local explanations picked up following the rule explained in subsection 4.3.2. As already mentioned, blue parts of the pictures have a negative effect on the output when excluded. Instead, when red parts are obscured, the predicted price raises. In even simpler words, blue windows are nice-to-have, red windows aren't. It can be seen that in most cases, the relevant windows are placed on the parts of the picture which regard the horse. Thus, the model recognises the horses as the main price drivers. In particular, the head, the neck and the front legs of the horses seem to have a positive impact on the predicted price, or at least, an impact which is relatively more positive than the one of the rear parts of the body (see Figure 20, 24, 28, 29, 30, 34 and 38). This may suggest that a close-up of the face of the animal is considered as a professional way to advertise it. In addition to this, the CNN seems to positively evaluate the presence of a rider in the picture (see Figure 19, 22, 23, 30, 31, 35 and 37). Perhaps, the rider is seen as a signal that the specimen has already been tamed and trained in a certain discipline, things which were found to be related to higher prices by previous models. It is also interesting to observe that when a picture includes two horses, the explanations get more confused, almost indicating that the model struggles to find relevant features (see Figure 19, 27 and 32). Moreover, in Figure 27 and 32, one of the two horses seems to drive down the price. Maybe, using a picture with two horses in the ad is confusing and unprofessional, thus it might only be done for cheap animals. Finally, there are a few explanations which are puzzling. Sometimes, elements of the background seem to be relevant price determinants. For instance, the sky seems to increase the predicted price in Figure 21. After all, the model's accuracy is not very high, and therefore it is not suprising that it occasionally relies on weird features. This reminds us that caution shall be used in leveraging on these insights for marketing practices. In fact, the predictions of the model have a low correlation with the real price of the specimen, hence features relevant for the model might not be that influencial in reality.

To sum up, the third unimodal model shows that the price of a horse can only be determined from an image to a limited extent. In fact, using images as an input leads to significantly worse results in prediction accuracy compared to the two previous models. Thus, horse traders are advised not to rely heavily on the picture when making price decisions. Once again, the saying "don't judge a book from its cover" is proven right. Nevertheless, the model-agnostic techniques enable us to understand the functioning of the model and derive some insights from it. In fact, model-agnostic explanations suggest that the head of a horse, the presence of a rider and the absence of a second horse prompt higher price predictions. This suggest that the sellers shall use a picture with a close-up of the face of the horse to professionally promote an expensive animal. They can also use a picture depicting a rider to effectively communicate that the animal is skilled and trained. Moreover, it may again suggest to the breeders that

tamed and trained animals are much more valuable. Still, every advice derived from these model-agnostic explanations needs to be treated with extreme caution, since the interpreted model is inaccurate.

5.4 Multimodal fusion model

The last model takes as input all the data at disposal. This means that this model uses the text description, the image and the other information contained in the horses' ads to predict their prices. This way, it can leverage on all the available pieces of information and their combination to increase the prediction's accuracy. Indeed, the different types of data may contain unique pieces of information. For instance, the image data may reveal some specific traits of the horse which were not described by the text or by the other variables. Moreover, combining together data with different modalities could lead to new relevant insights. The data input into the model are preprocessed exactly as described in the previous subsections. For instance, the same passages are followed to clean the text and to create the document-to-term matrix. Morever, the data are split into a training, a validation and a test set, exactly as done with the previous three models.

The data is input into the model through three input-branches, one for each data type. The structure of the input-branches is determined by the architecture of the optimal unimodal models. Thus, both the input-branch for categorical or numeric variables and the one for text data are composed by an input layer followed by two hidden layers with as many nodes as the ones used in the optimal unimodal models. The input-branch for images, instead, consists of only an input layer. In fact, it was found that it is not necessary to fine-tune the CNN, and thus image features extracted by the pre-trained convolutional blocks are fed into the branch. The layer concatenating the output of the input-branches has got 1853 nodes. Therefore, the number of nodes tested for the hidden layers following the concatenation layer are 232, 695, 1158 and 1621. These values are equally spaced in the interval [1, 1853]. In total, 180 different parameters' configurations are tested in the tuning process.

The minimum MSE over the validation set, reached by an architecture with no hidden layer in the decision head, is 0.4. This is an improvement on the best performing unimodal model, the one for text, which had an MSE of 0.44. The better perfomance indicates that the information contained in the different data types is supplementary. Indeed, if image features and numeric or categorical variables had not contained additional price determinants to the ones in the text data, the overall accuracy of the model would not have increased. Furthermore, the use of a hidden layer in the fusion head leads to further improvements in the performance, with 0.35 being the lowest validation loss observed. This might stem from the fact that the extra hidden layer allows for interactions between the data of the different modalities. Hence, it seems like the use of multimodal data and of interactions among them is beneficial.

The best performing model has a hidden layer with 1621 nodes and is trained with no decay, 0.001 as its learning rate and 32 as the batch-size. Figure 39 gives a graphical representation of the model. Its MSE on the test set is 0.39. A t-test comparing this to the MSE of the text model, rejects the null hypothesis with a p-value of 0.24. Hence, the latter MSE can be considered significantly better than the text one with a confidence interval of 0.76. Additionally, one-way analysis of variance reveals that the MSEs of all the four models are

statistically different with a p-value of 0. Figure 40 shows the plot of the residuals. The real-valued RMSE is \$4631, with an improvement of \$600 over previous results. The R^2 of the multimodal model is 0.58, the highest seen so far. Therefore, this last model achieves new state-of-the-art results in this field of research. This R^2 is almost twice as high as Freeborn's one (2009). Once again, Freeborn (2009) conducted a study similar to ours. She collected numeric and categorical data from online listings of horses. The gap between these results and hers highlights how important it is to have a multimodal approach to this problem.

To conclude, the multimodal fusion model outperforms all the three unimodal models on all the considered metrics. It does so by leveraging on the pieces of information contained in the different types of data. It seems to also benefit from the interactions between the different data. This implies that horse traders shall have a holistic approach to pricing decisions, considering the whole set of data given about the horse in different modalities. Concretely, the e-commerce website could provide a multimodal instrument to the horse sellers and buyers, to help them in determining the correct price for an animal. It is finally worth remembering that it is not possible to interpret this last model. Hence, unimodal models are still more suitable to give advice to sellers and buyers on how to better breed and promote horses.

6 Conclusion

The present thesis proposed to answer the following research question: *"to what extent do the multimodal features of a horse's ad on an online marketplace determine its price, when considering a deep learning approach?"*. The data necessary to conduct the research are web-scraped from an important e-commerce platform for horse trade, called *Equinenow*. In total, data from 6544 ads on the website are analysed. For each ad, numeric, categorical, text and image features are collected.

The data at hand are used to fit four different models. The first model takes as input only the numeric or categorical variables, the second only uses the text data, the third only uses the image data, while the fourth model exploits all the available data. The first three models are called "unimodal", since they only accept data in one modality. The fourth model is referred to as "multimodal fusion model", since it fuses together data of different modalities to make a prediction. Advanced machine learning techniques are employed to create the different models. In fact, all the models are neural networks, specifically designed to best suit the type of input data. The performance of every model is evaluated and compared. Moreover, for the first three models, model-agnostic techniques are applied to unveil if and in what way features influence the price prediction. Such a procedure cannot be done for the fourth model.

The text data unimodal model performs better than the model for numeric or categorical variables, which, in turn, outperforms the image model. In fact, the three models have a mean squared error over the test set of, respectively, 0.45, 0.55 and 0.78. The multimodal model further improves on the performance of the other regressions by reaching an MSE of 0.39. The four mean squared errors are proven to be significantly different by one-way analysis of variance. The fourth model has an R^2 of 0.58. It performs far better than the models from a previous study on the topic (Freeborn, 2009).

Model-agnostic techniques performed on unimodal models gave access to other interesting results. From the first unimodal model, it is found that the height and weight of a horse have a significant positive correlation with its price, and that age and temperament scores have a negative one. Moreover, standard ads seem to be related to lower prices than other more premium ads. Both the first and the second model produce several consistent results. In fact, they both highlight the importance of the breed in determining the price. For instance, andalusian and friesian horses appear to be more valuable than ponies, horses of mixed breeds or miniature ones. The horse's gender is found to be relevant by both models. For example, geldings seem to be more expensive than stallions, which, in turn, are more costly than mares. Having an ad with an embedded video or referring to a video in the text has a positive effect on the predicted price. Some horse's disciplines or skills are relevant price determinants as well. For example, it is appreciated if a horse is trained for western riding, jumping or if it is kid-safe. The second model doesn't only confirm the results of the first one, but it also provides new insights. In fact, it shows that more complete text descriptions are significantly associated with higher prices. Moreover, it reveals that words regarding contractual arrangements, horse's health and coat colour are important price drivers. For example, writing that a horse is injured or has worms decreases the price, while saying it is healthy raises it. Ads in which horse's boarding and hauling options are discussed generally have higher prices, and the opposite is true for the ads in which leasing agreements are

mentioned. Coat colours such as tobiano, buckskin and perlino are more valuable than dark ones. Finally, the third unimodal models shows that some image features might be relevant price determinants. For instance, images showing the face of a single horse and/or a rider seem to be preferable. However, these last insights shall be treated with caution, since the accuracy of the CNN is relatively low.

These results have several marketing implications. First, they are useful to breeders. Indeed, relevant relationships between certain genotypic and phenotypic characteristics of the animal and the price have been uncovered. This information can be employed to breed and train horses which better meet the market demand. For instance, cattle farmers shall rear the most remunerative breeds. Moreover, they shall geld their specimen, since stallions are less valued than geldings. They can also focus on training the horses in the skills and disciplines that are appreciated the most. At the same time, breeders could give less importance to the animal's features which are not found to have a relevant impact on the end price.

Second, many insights on how to correctly advertise a horse are unveiled. For example, it is proven that every part of the ad determines the price to a certain extent. However, it is shown that the text is the most important part of the ad, while image is the least important one. Thus, sellers shall spend more time on writing the text description than on taking pictures. In particular, longer text descriptions are appreciated. Including a video in the ad appears to be another winning promotion strategy. Furthermore, the study reveals which words are important and beneficial to use in the ad. The sellers could optimise their ads by using these words. Perhaps, the website could even develop a promotion tool able to give suggestions to sellers on improving the animal's description. Similarly, guidelines on how to pick the best picture for the ad could be given to the seller based on the image features which are found to be most relevant.

Third, the developed models have proven effective in solving the pricing problem. This result is even more important considering that the examined market is characterised by internationtional, peer-to-peer, online transactions of unique, luxury goods. Moreover, the multimodal model shows that it is best to take a holistic approach to pricing horses. Hence, horse traders are advised to consider all the data at their disposal to determine the value of the specimen. However, relatively less importance shall be given to images when taking price decisions. Many price drivers are individuated as well, which could provide guidance on how to correctly price a horse to inexperienced traders. Furthermore, *Equinenow* would probably benefit from creating a pricing tool based on the multimodal model, given its good performances. Such a tool would suggest fair prices to website's users, both to buyers and sellers. This would make the market more transparent and would foster market clearing.

Fourth, given the encouraging results, the approach used in this research could be applied to other, similar problems. It could be beneficial in many other trading contexts, including online, peer-to-peer, good-specific, luxury and international trades. This may sound unimportant, but it is not. E-commerce will represent 22% of global retail sales by 2023 (eMarketer, 2019). Moreover, in Europe, 20% of the population sells products on peer-to-peer online marketplaces (Eurostat, 2020). The listings on most of these platforms include text and pictures, besides structured data. Therefore, there is a big number of websites that can benefit from the approach outlined in this thesis. To give a concrete example, Marktplaats could create

similar models to analyse the second-hand bikes' market. The website would gain a better understanding of this trade and would be able to improve the users' experience with some new online tools. To conclude, the relevance of this thesis is not limited to the horse-market context.

Given the interesting results and their numerous implications, the method used in the thesis can be said effective. This gives academic relevance to the thesis. The study contributes to marketing literature by bringing together for the first time many advanced data science techniques and a well-known piece of economic theory. Indeed, there are many studies based on the hedonic pricing theory, but few of them used machine learning methods. Moreover, extensive research was done on multimodal models, but they were rarely used for pricing purposes. The literature on computer vision methods is broad, but it rarely treats pricing problems. This thesis, instead, uses all these methods. Moreover, it demonstrates that these complex machine learning models can still be interpreted, thanks to model-agnostic techniques. Therefore, hedonic pricing models shall not be limited to simple linear regressions but shall use more advanced techniques to improve the results and include more data. To conclude, this thesis brings together an old economic theory and three of the most advanced field of research in machine learning. The results demonstrate that such a union is highly beneficial.

However, this thesis, as any other research, is subjected to some limitations. Certain limitations stem from the data that are employed. Since the data are taken from a peer-to-peer website, they are imprecise sometimes. For instance, some website users do not fill in all the fields of the ad, while others fill them incorrectly. Moreover, the number of samples collected is six thousand. This is higher than the number used in previous studies about hedonic pricing in the horse-market, but it is far lower than the number usually employed to train CNNs. Nevertheless, transfer learning allows us to exploit the image data, even if the training set is small. Furthermore, the data collected regard ads present on the website in a certain time interval. No ad stays on the website for more than three months. Hence, the relevance of the findings could be limited to a certain time span. If any of the models were employed to build website tools, a periodic retraining of the algorithms would be advisable. Finally, the prices posted by the sellers could potentially be changed during negotiations with the buyer or could not be met by the demand. However, previous research has found that the end price is often similar to the posted one (Freeborn, 2009). Overall, the data limitations seem not to affect the study too much. Other limitations are linked to the hardware at our disposal. The scope of the architecture search and of the hyperparameter tuning process was often limited by the necessity not to overflood the RAM and to respect time constraints. Finally, the method used is very sophisticated, but there is room to improve it even further. For instance, a feed-forward neural network is used for the text data, but there are more advanced text-analytics algorithms. In fact, the state-of-the-art techniques for natural language processing are recurrent neural networks. These algorithms could perhaps reach even better performances, but they would also be more difficult to interpret.

This thesis may inspire future research. This study's approach to hedonic pricing in the horse-market is exhaustive. Thus, no further research is needed on the topic. However, the same approach could be applied to different e-commerce platforms selling different goods, in the attempt to highlight differences and similarities in marketing practices among different domains. This could lead to interesting results. Moreover, the method could be further

developed by the use of more advanced natural language processing neural networks. This advancement shall be done together with an exploration of model-agnostic techniques able to interpret the results of recurrent neural networks. This way, the tradeoff between accuracy and interpretability would be more convenient. Model-agnostic techniques for regression over image data shall be further researched as well. In fact, even if some insights could be mined about the functioning of the CNN in this thesis, the interpretation process was more complicated and less reliable than the one used for the first two unimodal models.

To conclude, a multimodal approach to hedonic pricing for the peer-to-peer e-commerce of horses was developed and succesfully applied to web-scraped data. The results are relevant and interesting, and lead to several marketing implications. Some limitations and possible future research directions were discussed as well.

References

Alom, M. Z., Taha, T. M., Yakopcic, C., Westberg, S., Sidike, P., Nasrin, M. S., Van Essen, B.C., Awwal, A. A. S., & Asari, V. K. (2018). The history began from alexnet: A comprehensive survey on deep learning approaches. *arXiv preprint arXiv:1803.01164*.

Ahmed, E. H., & Moustafa, M. (2016). House Price Estimation from Visual and Textual Features. *Proceedings of the 8th International Joint Conference on Computational Intelligence.* doi: 10.5220/0006040700620068

Atrey, P. K., Hossain, M. A., El Saddik, A., & Kankanhalli, M. S. (2010). Multimodal fusion for multimedia analysis: a survey. *Multimedia systems*, 16(6), 345-379.

Baltrušaitis, T., Ahuja, C., & Morency, L. P. (2018). Multimodal machine learning: A survey and taxonomy. *IEEE transactions on pattern analysis and machine intelligence*, 41(2), 423-443.

Bishop, C. M. (1995). Neural networks for pattern recognition. Oxford university press.

Bonnett, C. (2016, June 26). Classifying e-commerce products based on images and text. Retrieved from https://cbonnett.github.io/Insight.html

Chen, S., Chou, E., & Yang, R. (2018). The Price is Right: Predicting Prices with Product Images. *arXiv preprint arXiv*.

Deng, J., Dong, W., Socher, R., Li, L. J., Li, K., & Fei-Fei, L. (2009, June). Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition* (pp. 248-255). Ieee.

Donnet, M. L., Weatherspoon, D. D., & Hoehn, J. P. (2007). What adds value in specialty coffee? Managerial implications from hedonic price analysis of Central and South American e-auctions. *International Food and Agribusiness Management Review*, 10(1030-2016-82517), 1-18.

Douglas, J. (March 19, 2007). World's smallest horse has tall order. *The Washington Post. Associated Press*. Retrieved from https://www.washingtonpost.com

eMarketer. (June 27, 2019). E-commerce share of total global retail sales from 2015 to 2023 [Graph]. In *Statista*. Retrieved August 07, 2020, from https://www.statista.com/statistics/534123/e-commerce-share-of-retail-sales-worldwide/

Equine Business Association. (n.d.). Equine Industry Statistics Overview. Retrieved from https://www.equinebusinessassociation.com/equine-industry-statistics/

Eurostat. (April 15, 2020) Individuals - internet activities. [Data file]. Retrieved from https://ec.europa.eu/eurostat

Freeborn, J. (2009). *Hedonic price analysis of the internet recreational equine market* (Doctoral dissertation, Kansas State University).

Friedman, J., Hastie, T., & Tibshirani, R. (2001). *The elements of statistical learning* (Vol. 1, No. 10). New York: Springer series in statistics.

Gille, C., Kayser, M., & Spiller, A. (2010). Target Group Segmentation in the Horse Buyers' Market against the Background of Equestrian Experience. *Journal of equine science*, 21(4), 67-71.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT press.

Guide to First-Time Horse Ownership. (n.d.). Retrieved from https://extension.umaine.edu/publicationns/1004e/

Heaton, J. (2008). *Introduction to neural networks with Java*. Heaton Research, Inc..

Hopkinson, G. C., & Pujari, D. (1999). A factor analytic study of the sources of meaning in hedonic consumption. *European Journal of Marketing*.

Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., & Adam, H. (2017). Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*.

Hunter, A. (2003). *American Classic Pedigrees (1914-2002)*. Eclipse Press.

Iandola, F. N., Han, S., Moskewicz, M. W., Ashraf, K., Dally, W. J., & Keutzer, K. (2016). SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and< 0.5 MB model size. *arXiv preprint arXiv:1602.07360*.

Joo, J., & Steinert-Threlkeld, Z. C. (2018). Image as data: Automated visual content analysis for political science. *arXiv preprint arXiv:1810.01544*.

Keskar, N. S., Mudigere, D., Nocedal, J., Smelyanskiy, M., & Tang, P. T. P. (2016). On large-batch training for deep learning: Generalization gap and sharp minima. *arXiv preprint arXiv:1609.04836*.

Kornblith, S., Shlens, J., & Le, Q. V. (2019). Do better imagenet models transfer better?. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2661-2671).

Lancaster, K. J. (1966). A new approach to consumer theory. *Journal of political economy*, 74(2), 132-157.

Lange, K. Y., Johnson, J. W., Wilson, K., & Johnson, W. (2010). *Price Determinants of Ranch Horses Sold at Auction in Texas* (No. 1370-2016-108774).

Lansford Jr, N. H., Freeman, D. W., Topliff, D. R., & Walker, O. L. (1998). Hedonic pricing of race-bred yearling Quarter Horses produced by Quarter Horse sires and dams. *Journal of Agribusiness*, 16(2), 1-17.

Lei, S., Zhang, H., Wang, K., & Su, Z. (2018). How Training Data Affect the Accuracy and Robustness of Neural Networks for Image Classification.

Maynard, L. J., & Stoeppel, K. M. (2007). Hedonic price analysis of thoroughbred broodmares in foal. *Journal of Agribusiness*, 25(345-2016-15149), 181-195.

McCarthy, E. J. (1964). *Basic Marketing*, IL: Richard D. Irwin.

Minaee, S., Kalchbrenner, N., Cambria, E., Nikzad, N., Chenaghlu, M., & Gao, J. (2020). Deep learning based text classification: A comprehensive review. *-arXiv preprint arXiv:2004.03705*.

Meier, A. (2013, March 10). Morbid Monday: The Split Head of Old Billy, the World's Oldest Horse. Retrieved from https://www.atlasobscura.com/articles/morbid-monday-split-head-of-the-worlds-oldest-horse

Molnar, C. (2019). *Interpretable machine learning*. Lulu. com.

Oddie, C. F., Hawson, L. A., McLean, A. N., & McGreevy, P. D. (2014). Do vendors value safety in the Australian recreational (non-thoroughbred) riding horse market?. *Journal of Veterinary Behavior*, 9(6), 375-381.

Ortega, J. D., Senoussaoui, M., Granger, E., Pedersoli, M., Cardinal, P., & Koerich, A. L. (2019). Multimodal fusion with deep neural networks for audio-video emotion recognition. *arXiv preprint arXiv:1907.03196*.

Reed, R., & Marks, R. J. II (1999). *Neural smithing: supervised learning in feedforward artificial neural networks*. Mit Press.

Ribeiro, M. T., Singh, S., & Guestrin, C. (2016, August). " Why should i trust you?" Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 1135-1144).

Rosen, S. (1974). Hedonic prices and implicit markets: product differentiation in pure competition. *Journal of political economy*, 82(1), 34-55.

Rosebrock, A. (2019, February 4). Keras: Multiple Inputs and Mixed Data. Retrieved from https://www.pyimagesearch.com/2019/02/04/keras-multiple-inputs-and-mixed-data/

Ruder, S. (2016). An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*.

Saeed, K., & Snášel, V. (Eds.). (2014). *Computer Information Systems and Industrial Management: 13th IFIP TC 8 International Conference, CISIM 2014, Ho Chi Minh City, Vietnam, November 5-7, 2014, Proceedings* (Vol. 8838). Springer.

Sirmans, S., Macpherson, D., & Zietz, E. (2005). The composition of hedonic pricing models. *Journal of real estate literature*, 13(1), 1-44.

Steiner, B. E. (2004). Australian wines in the British wine market: a hedonic price analysis. *Agribusiness: An International Journal*, 20(3), 287-307.

Stekhoven, D. J., & Bühlmann, P. (2012). MissForest—non-parametric missing value imputation for mixed-type data. *Bioinformatics*, 28(1), 112-118.

Sun, Y., Zhu, L., Wang, G., & Zhao, F. (2017). Multi-input convolutional neural network for flower grading. *Journal of Electrical and Computer Engineering*, 2017.

Tan, M., & Le, Q. V. (2019). Efficientnet: Rethinking model scaling for convolutional neural networks. *arXiv preprint arXiv:1905.11946*.

Taylor, M. R., Dhuyvetter, K. C., Kastens, T. L., Douthit, M., & Marsh, T. L. (2006). Show quality quarter horse auctions: price determinants and buy-back practices. *Journal of Agricultural and Resource Economics*, 595-615.

Yosinski, J., Clune, J., Bengio, Y., & Lipson, H. (2014). How transferable are features in deep neural networks?. *In Advances in neural information processing systems* (pp. 3320-3328).

Zeiler, M. D., & Fergus, R. (2014, September). Visualizing and understanding convolutional networks. In *European conference on computer vision* (pp. 818-833). Springer, Cham.

Zhang, K., & Zhang, K. (2019). PetFinder Challenge: Predicting Pet Adoption Speed.

Tables

Table 2: Feature importance permutation results for the first unimodal model.

Feature	*importance.05*	*importance*	*importance.95*	*permutation.error*
Weight	0.939	1.059	1.137	0.580
FoalDate	0.954	1.042	1.179	0.572
hasVideo	0.947	1.038	1.162	0.569
typeOfAdstandard.photo.ad	0.932	1.035	1.161	0.568
Breedgypsy.vanner	0.937	1.033	1.140	0.567
Height	0.878	1.033	1.164	0.566
Breedfriesian	0.929	1.029	1.118	0.564
Breedandalusian	0.932	1.028	1.149	0.564
disciplinekidsafe	0.951	1.027	1.121	0.563
Breedhanoverian	0.905	1.027	1.170	0.563
StateBredkansas	0.903	1.026	1.117	0.563
Registryghra	0.913	1.025	1.119	0.562
Registrybwp	0.882	1.024	1.123	0.562
disciplineallaround	0.925	1.024	1.130	0.561
Gendergelding	0.924	1.023	1.122	0.561
Markingsblaze	0.904	1.023	1.115	0.561
Temperament	0.918	1.022	1.125	0.561
StateBrednew.york	0.910	1.022	1.114	0.560
Registryrmha	0.928	1.021	1.142	0.560
disciplineenglishpleasure	0.912	1.020	1.143	0.559
StateBredwest.virginia	0.894	1.020	1.121	0.559
locationmichigan	0.890	1.020	1.147	0.559
Breedpinto	0.912	1.020	1.104	0.559
Registrycfha	0.910	1.019	1.116	0.559
Genderunborn.foal	0.902	1.018	1.151	0.558
hasPedigree	0.889	1.017	1.086	0.558
Registrygov	0.902	1.017	1.103	0.558
Colorwhite	0.899	1.017	1.109	0.557
locationohio	0.923	1.017	1.118	0.557
StateBredgeorgia	0.932	1.016	1.102	0.557
Breedpaint.pony	0.918	1.016	1.096	0.557
Breedappendix	0.928	1.016	1.139	0.557
disciplinehusbandsafe	0.911	1.016	1.161	0.557
locationmaryland	0.913	1.015	1.125	0.556
Breedtrakehner	0.879	1.014	1.123	0.556
Registrymfthba	0.911	1.014	1.137	0.556
Registryapha	0.906	1.014	1.110	0.556
Genderunknown	0.915	1.014	1.131	0.556
Registryancce	0.904	1.013	1.133	0.556

Feature	*importance.05*	*importance*	*importance.95*	*permutation.error*
Breedminiature	0.897	1.013	1.141	0.555
Registryhahr	0.942	1.013	1.132	0.555
disciplinenone	0.923	1.013	1.140	0.555
Genderfilly	0.917	1.013	1.147	0.555
Breedoldenburg	0.937	1.013	1.179	0.555
Breedarabian	0.923	1.012	1.117	0.555
StateBredkentucky	0.906	1.012	1.112	0.555
Breedlusitano	0.905	1.011	1.147	0.554
Colorred.dun	0.920	1.011	1.114	0.554
locationsaskatchewan	0.883	1.011	1.124	0.554
StateBredsaskatchewan	0.888	1.011	1.124	0.554
locationconnecticut	0.913	1.011	1.121	0.554
Breedpalomino	0.883	1.010	1.129	0.554
Breedmorgan	0.918	1.010	1.120	0.554
Registryasha	0.917	1.010	1.138	0.554
Colorovero	0.920	1.010	1.150	0.554
StateBredutah	0.901	1.010	1.117	0.554
disciplineeventing	0.905	1.010	1.122	0.554
disciplinedressage	0.893	1.010	1.112	0.554
Breedazteca	0.915	1.010	1.079	0.554
Registryrpsi	0.912	1.010	1.122	0.554
locationcalifornia	0.917	1.010	1.114	0.554
locationbritish.columbia	0.908	1.010	1.127	0.554
Registrykfps	0.909	1.010	1.128	0.554
Breedholsteiner	0.904	1.010	1.115	0.554
Colorother	0.901	1.010	1.101	0.554
Registryamhr	0.902	1.010	1.144	0.554
Registryoldna	0.910	1.009	1.107	0.553
StateBredbritish.columbia	0.904	1.009	1.104	0.553
Colorcremello	0.897	1.009	1.118	0.553
Genderweanling	0.902	1.009	1.102	0.553
disciplineflashy	0.902	1.009	1.133	0.553
disciplinejumping	0.918	1.009	1.105	0.553
locationillinois	0.883	1.009	1.138	0.553
Breedtennessee.walking	0.922	1.009	1.116	0.553
locationmassachusetts	0.911	1.009	1.139	0.553
disciplineranch	0.912	1.009	1.139	0.553
locationarizona	0.912	1.008	1.105	0.553
Colorliver.chestnut	0.919	1.007	1.110	0.552
Colorgrey	0.918	1.007	1.099	0.552
Colorchampagne	0.886	1.007	1.109	0.552
locationutah	0.924	1.006	1.106	0.552
Registryaahia	0.906	1.006	1.106	0.552
locationmissouri	0.912	1.006	1.114	0.552

Feature	*importance.05*	*importance*	*importance.95*	*permutation.error*
disciplinelesson	0.901	1.006	1.103	0.552
locationwashington	0.906	1.006	1.084	0.552
Registryaha	0.896	1.006	1.150	0.551
Registrygvhs	0.883	1.006	1.130	0.551
Registryahr	0.888	1.005	1.127	0.551
Breedthoroughbred	0.928	1.005	1.112	0.551
Colordun	0.897	1.005	1.094	0.551
StateBrednew.mexico	0.893	1.005	1.103	0.551
Colorbay.overo	0.929	1.004	1.102	0.551
currencyusd	0.900	1.004	1.098	0.551
Breeddutch.warmblood	0.939	1.004	1.139	0.551
locationnew.mexico	0.891	1.004	1.095	0.551
Registrycwhba	0.894	1.004	1.118	0.551
locationvirginia	0.912	1.004	1.108	0.551
Breedquarter.horse	0.904	1.004	1.113	0.551
InFoal	0.889	1.004	1.119	0.551
StateBredmontana	0.909	1.004	1.128	0.550
Registrycahr	0.876	1.004	1.126	0.550
Colorbuckskin	0.913	1.004	1.106	0.550
StateBredoklahoma	0.910	1.004	1.123	0.550
StateBrediowa	0.892	1.003	1.076	0.550
locationidaho	0.915	1.003	1.122	0.550
Colorroan	0.897	1.003	1.117	0.550
locationgeorgia	0.918	1.003	1.117	0.550
StateBredmichigan	0.905	1.003	1.083	0.550
Registryno.registration	0.912	1.003	1.102	0.550
locationnew.york	0.901	1.002	1.096	0.550
locationminnesota	0.889	1.002	1.081	0.550
Breedwestphalian	0.905	1.002	1.109	0.550
Registryusihc	0.930	1.002	1.144	0.550
Breedclydesdale	0.911	1.002	1.094	0.549
Breedshetland.pony	0.899	1.002	1.128	0.549
locationquebec	0.886	1.002	1.126	0.549
StateBredwyoming	0.892	1.002	1.101	0.549
locationmontana	0.922	1.001	1.103	0.549
Colorchocolate	0.890	1.001	1.121	0.549
StateBredsouth.dakota	0.872	1.001	1.120	0.549
disciplinebeginner	0.913	1.001	1.116	0.549
locationarkansas	0.877	1.001	1.134	0.549
Gendercolt	0.894	1.001	1.085	0.549
Registrycths	0.893	1.001	1.155	0.549
Markingsstock	0.920	1.001	1.164	0.549

Table 3: Feature importance permutation results for the second unimodal model.

feature	importance.05	importance	importance.95	permutation.error
fanci	0.975	1.109	1.178	0.500
jump	0.955	1.103	1.146	0.498
break	0.954	1.097	1.147	0.495
stride	0.913	1.091	1.149	0.493
around	0.972	1.091	1.192	0.492
cattl	0.873	1.089	1.257	0.491
board	0.954	1.085	1.208	0.490
tripl	0.898	1.085	1.257	0.490
train	0.917	1.084	1.176	0.489
poni	0.978	1.083	1.172	0.489
healthi	0.957	1.081	1.166	0.488
age	0.922	1.071	1.131	0.483
grade	0.964	1.069	1.154	0.483
confid	0.951	1.069	1.139	0.482
worm	0.941	1.069	1.193	0.482
mare	0.889	1.068	1.110	0.482
'll	0.935	1.067	1.211	0.481
pleas	0.878	1.066	1.145	0.481
pay	0.914	1.066	1.174	0.481
face	0.899	1.065	1.163	0.481
leas	0.921	1.065	1.241	0.481
unfortun	0.925	1.065	1.152	0.480
abl	0.913	1.063	1.132	0.480
warmblood	0.943	1.063	1.190	0.480
halter	0.913	1.062	1.168	0.479
sold	0.921	1.062	1.182	0.479
water	0.887	1.061	1.171	0.479
man	0.980	1.061	1.154	0.479
tobiano	0.934	1.060	1.212	0.479
friesian	0.920	1.058	1.200	0.477
spanish	0.959	1.058	1.158	0.477
equin	0.880	1.057	1.118	0.477
real	0.910	1.057	1.161	0.477
video	0.970	1.057	1.160	0.477
injuri	0.921	1.056	1.103	0.477
numberOfWords	0.923	1.056	1.180	0.476
alisha	0.882	1.055	1.123	0.476
snaffl	0.883	1.055	1.099	0.476
advanc	0.909	1.054	1.122	0.475
deworm	0.958	1.054	1.087	0.475
andalusian	0.880	1.053	1.156	0.475

feature	importance.05	importance	importance.95	permutation.error
buckskin	0.861	1.052	1.223	0.475
need	0.896	1.052	1.129	0.475
perlino	0.929	1.052	1.109	0.475
fall	0.907	1.051	1.122	0.474
hind	0.924	1.051	1.136	0.474
deserv	0.858	1.050	1.189	0.474
countri	0.933	1.050	1.164	0.474
plenti	0.932	1.049	1.118	0.474
hot	0.947	1.049	1.112	0.473
serious	0.907	1.049	1.120	0.473
varieti	0.880	1.049	1.099	0.473
miniatur	0.972	1.048	1.201	0.473
amha	0.920	1.048	1.170	0.473
basic	0.905	1.048	1.134	0.473
pretti	0.944	1.048	1.146	0.473
left	0.935	1.048	1.172	0.473
haul	0.972	1.046	1.148	0.472
dark	0.896	1.046	1.175	0.472
realli	0.893	1.046	1.114	0.472
dream	0.935	1.046	1.092	0.472
can	0.951	1.046	1.163	0.472
foundat	0.945	1.045	1.118	0.472
west	0.952	1.045	1.147	0.472
track	0.954	1.045	1.131	0.472
tune	0.923	1.045	1.149	0.471
desir	0.915	1.044	1.204	0.471
forev	0.919	1.044	1.130	0.471
spain	0.923	1.044	1.169	0.471
other	0.888	1.044	1.230	0.471
amhr	0.930	1.043	1.137	0.471
given	0.932	1.043	1.107	0.471
firm	0.935	1.043	1.188	0.471
har	0.987	1.043	1.166	0.471
gorgeous	0.992	1.043	1.110	0.471
throughout	0.829	1.043	1.098	0.471
sporthors	0.906	1.043	1.113	0.471
text	0.833	1.043	1.145	0.470
activ	0.913	1.043	1.118	0.470
movement	0.868	1.042	1.208	0.470
appaloosa	0.884	1.042	1.106	0.470
dead	0.926	1.041	1.136	0.470
free	0.945	1.041	1.145	0.470
rear	0.929	1.041	1.129	0.470
daili	0.847	1.041	1.105	0.470

feature	importance.05	importance	importance.95	permutation.error
throw	0.944	1.040	1.116	0.469
pull	0.949	1.040	1.137	0.469
qualiti	0.947	1.040	1.197	0.469
sibl	0.884	1.040	1.190	0.469
husband	0.883	1.040	1.103	0.469
true	0.938	1.040	1.186	0.469
parad	0.916	1.040	1.156	0.469
pictur	0.905	1.039	1.119	0.469
email	0.923	1.039	1.223	0.469
book	0.953	1.039	1.121	0.469
feel	0.903	1.039	1.173	0.469
hoof	0.914	1.039	1.147	0.469
stay	0.985	1.039	1.143	0.469
easi	0.954	1.039	1.193	0.469
athlet	0.977	1.038	1.176	0.469
gold	0.869	1.038	1.116	0.468
bite	0.927	1.038	1.105	0.468
san	0.884	1.038	1.140	0.468
schedul	0.863	1.037	1.126	0.468
team	0.837	1.037	1.164	0.468
life	0.935	1.037	1.176	0.468
conform	0.937	1.036	1.094	0.468
last	0.940	1.036	1.128	0.468
stud	0.904	1.036	1.171	0.468
inspect	0.957	1.036	1.179	0.468
buck	0.911	1.036	1.159	0.467
pssm	0.882	1.036	1.133	0.467
group	0.904	1.036	1.200	0.467
speed	0.959	1.035	1.134	0.467
almost	0.948	1.035	1.125	0.467
pen	0.931	1.035	1.184	0.467
hay	0.900	1.034	1.146	0.467
tack	0.976	1.034	1.144	0.466
consist	0.953	1.034	1.130	0.466
inquir	0.901	1.034	1.119	0.466
solid	0.844	1.034	1.101	0.466
chang	0.953	1.033	1.098	0.466
far	0.937	1.033	1.144	0.466
clinic	0.944	1.033	1.112	0.466
away	0.863	1.033	1.139	0.466
facil	0.882	1.033	1.155	0.466
strong	0.887	1.033	1.143	0.466
bareback	0.906	1.032	1.122	0.466
payment	0.888	1.032	1.140	0.466

feature	importance.05	importance	importance.95	permutation.error
hes	0.926	1.032	1.084	0.466
shoe	0.848	1.032	1.160	0.466
natur	0.926	1.032	1.164	0.466
reg	0.910	1.032	1.131	0.465
apha	0.861	1.032	1.153	0.465
ground	0.907	1.031	1.091	0.465
larg	0.892	1.031	1.120	0.465
straight	0.884	1.031	1.209	0.465
youth	0.903	1.031	1.076	0.465
purchas	0.908	1.031	1.151	0.465
bad	0.918	1.031	1.171	0.465
trust	0.957	1.030	1.056	0.465
championship	0.915	1.030	1.146	0.465
sane	0.902	1.030	1.175	0.465
pedigre	0.956	1.030	1.098	0.465
fulli	0.949	1.030	1.098	0.465
mount	0.935	1.029	1.153	0.465
chocol	0.933	1.029	1.151	0.464
associ	0.909	1.029	1.109	0.464
career	0.981	1.029	1.110	0.464
money	0.857	1.029	1.116	0.464
mean	0.952	1.029	1.180	0.464
grulla	0.889	1.029	1.080	0.464
partner	0.938	1.029	1.078	0.464
offspr	0.908	1.029	1.081	0.464
person	0.946	1.029	1.147	0.464
usa	0.942	1.029	1.092	0.464
miss	0.905	1.029	1.087	0.464
feed	0.953	1.028	1.079	0.464
rack	0.874	1.028	1.172	0.464
florida	1.003	1.028	1.187	0.464
farrier	0.962	1.028	1.157	0.464
cowboy	0.911	1.028	1.179	0.464
equestrian	0.882	1.027	1.165	0.464
horsemanship	0.925	1.027	1.124	0.464
roan	0.927	1.027	1.125	0.463
wtc	0.906	1.027	1.132	0.463
page	0.950	1.027	1.135	0.463
nation	0.943	1.027	1.098	0.463
incred	0.925	1.027	1.158	0.463
along	0.929	1.027	1.148	0.463
yet	0.961	1.026	1.092	0.463
ranch	0.901	1.026	1.174	0.463
round	0.897	1.026	1.157	0.463

feature	importance.05	importance	importance.95	permutation.error
bit	0.916	1.026	1.100	0.463
approv	0.905	1.026	1.166	0.463
set	0.894	1.026	1.118	0.463
size	0.938	1.026	1.157	0.463
purebr	0.933	1.026	1.101	0.463
thing	0.951	1.026	1.104	0.463
cant	0.922	1.026	1.190	0.463
tall	0.916	1.026	1.086	0.463
east	0.954	1.025	1.109	0.463
donkey	0.886	1.025	1.131	0.463
lifetim	0.962	1.025	1.093	0.463
golden	0.864	1.025	1.093	0.463
get	0.821	1.025	1.125	0.463
measur	0.937	1.025	1.049	0.463
first	0.927	1.025	1.167	0.462
log	0.884	1.025	1.122	0.462
wonder	0.903	1.025	1.109	0.462
shod	0.917	1.025	1.131	0.462
clip	0.926	1.024	1.079	0.462
march	0.955	1.024	1.120	0.462
four	0.895	1.024	1.127	0.462
love	0.922	1.024	1.176	0.462
educ	0.929	1.024	1.137	0.462
weight	0.970	1.023	1.174	0.462
point	0.973	1.023	1.256	0.462
groundwork	0.937	1.023	1.120	0.462
minut	0.867	1.023	1.231	0.462
long	0.932	1.023	1.128	0.462
everi	0.939	1.023	1.088	0.461
ring	0.960	1.022	1.112	0.461
super	0.941	1.022	1.240	0.461
enough	0.917	1.022	1.103	0.461
human	0.862	1.022	1.161	0.461
bath	0.858	1.022	1.141	0.461
without	0.927	1.022	1.075	0.461
transit	0.928	1.021	1.129	0.461
meet	0.934	1.021	1.150	0.461
reserv	0.849	1.021	1.209	0.461
oper	0.991	1.021	1.059	0.461
bolt	0.899	1.021	1.188	0.461
barrel	0.941	1.021	1.089	0.461
hoov	0.882	1.021	1.181	0.461
kick	0.903	1.021	1.120	0.461
balanc	0.924	1.021	1.119	0.461

feature	importance.05	importance	importance.95	permutation.error
winner	0.904	1.021	1.111	0.461
saddl	0.900	1.021	1.078	0.460
form	0.946	1.020	1.067	0.460
pace	0.929	1.020	1.101	0.460
aqha	0.933	1.020	1.094	0.460
youtub	0.911	1.020	1.180	0.460
someth	0.946	1.020	1.108	0.460

Table 4: Feature effect for the most important text features.

	Feature	*Importance*	*Effect*		*Feature*	*Importance*	*Effect*
1	fanci	1.109	-0.005	42	need	1.052	-0.371
2	jump	1.103	0.175	43	perlino	1.052	0.099
3	break.	1.097	0.000	44	fall	1.051	-0.105
4	stride	1.091	0.117	45	hind	1.051	-0.219
5	around	1.091	-0.209	46	deserv	1.050	-0.113
6	cattl	1.089	0.230	47	countri	1.050	-0.005
7	board	1.085	-0.396	48	plenti	1.049	-0.078
8	tripl	1.085	0.084	49	hot	1.049	-0.036
9	train	1.084	0.191	50	serious	1.049	0.051
10	poni	1.083	-0.829	51	varieti	1.049	0.144
11	healthi	1.081	0.084	52	miniatur	1.048	-1.236
12	age	1.071	-0.200	53	amha	1.048	-0.206
13	grade	1.069	-0.962	54	basic	1.048	-0.029
14	confid	1.069	-0.015	55	pretti	1.048	-0.079
15	worm	1.069	-0.051	56	left	1.048	-0.251
16	mare	1.068	-0.139	57	haul	1.046	0.510
17	’ll	1.067	-0.064	58	dark	1.046	-0.018
18	pleas	1.066	0.230	59	realli	1.046	0.069
19	pay	1.066	0.185	60	dream	1.046	0.100
20	face	1.065	0.177	61	can	1.046	-0.156
21	leas	1.065	-2.071	62	foundat	1.045	0.097
22	unfortun	1.065	-0.079	63	west	1.045	0.093
23	abl	1.063	-0.237	64	track	1.045	0.170
24	warmblood	1.063	0.643	65	tune	1.045	0.101
25	halter	1.062	-0.995	66	desir	1.044	0.122
26	sold	1.062	-0.323	67	forev	1.044	-0.105
27	water	1.061	0.071	68	spain	1.044	0.508
28	man	1.061	-0.319	69	other	1.044	0.455
29	tobiano	1.060	0.171	70	amhr	1.043	-0.539
30	friesian	1.058	1.205	71	given	1.043	-0.178
31	spanish	1.058	0.171	72	firm	1.043	-0.021
32	equin	1.057	-0.293	73	har	1.043	0.057
33	real	1.057	0.228	74	gorgeous	1.043	-0.002
34	video	1.057	0.496	75	throughout	1.043	0.079
35	injuri	1.056	-0.244	76	sporthors	1.043	0.062
36	alisha	1.055	0.072	77	text	1.043	-0.127
37	snaffl	1.055	0.008	78	activ	1.043	-0.118
38	advanc	1.054	0.004	79	movement	1.042	0.358
39	deworm	1.054	0.006	80	appaloosa	1.042	0.057
40	andalusian	1.053	1.391	81	dead	1.041	-0.023
41	buckskin	1.052	0.178	82	free	1.041	-0.256

Figures

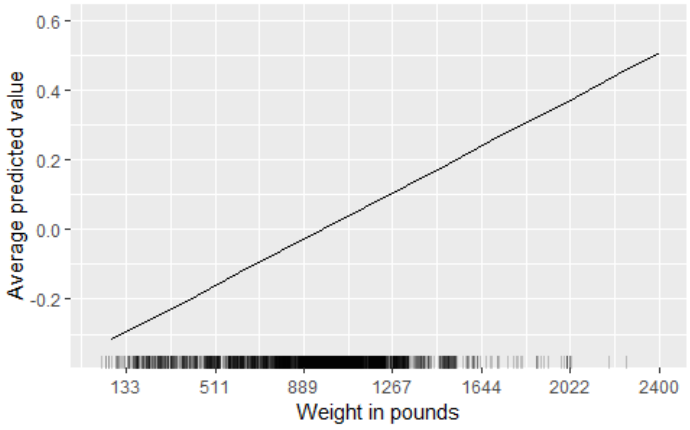

Figure 4: Partial dependence plot of the weight's effect over prediction.

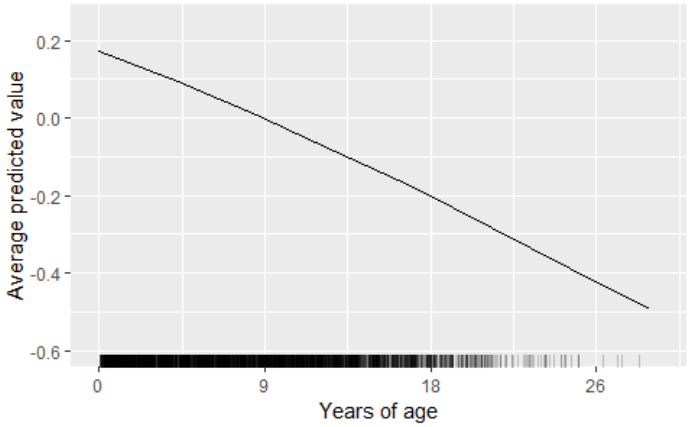

Figure 5: Partial dependence plot of the foal date's effect over prediction.

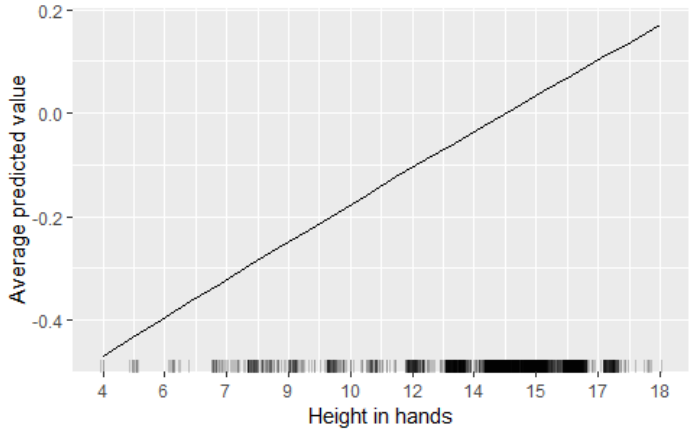

Figure 6: Partial dependence plot of the height's effect over prediction.

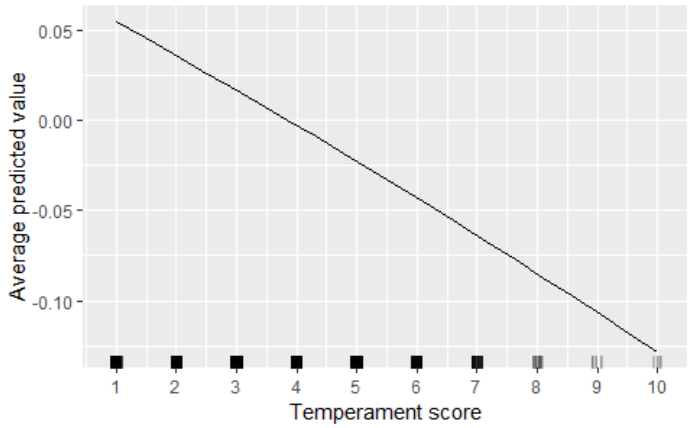

Figure 7: Partial dependence plot of the temperament score's effect over prediction.

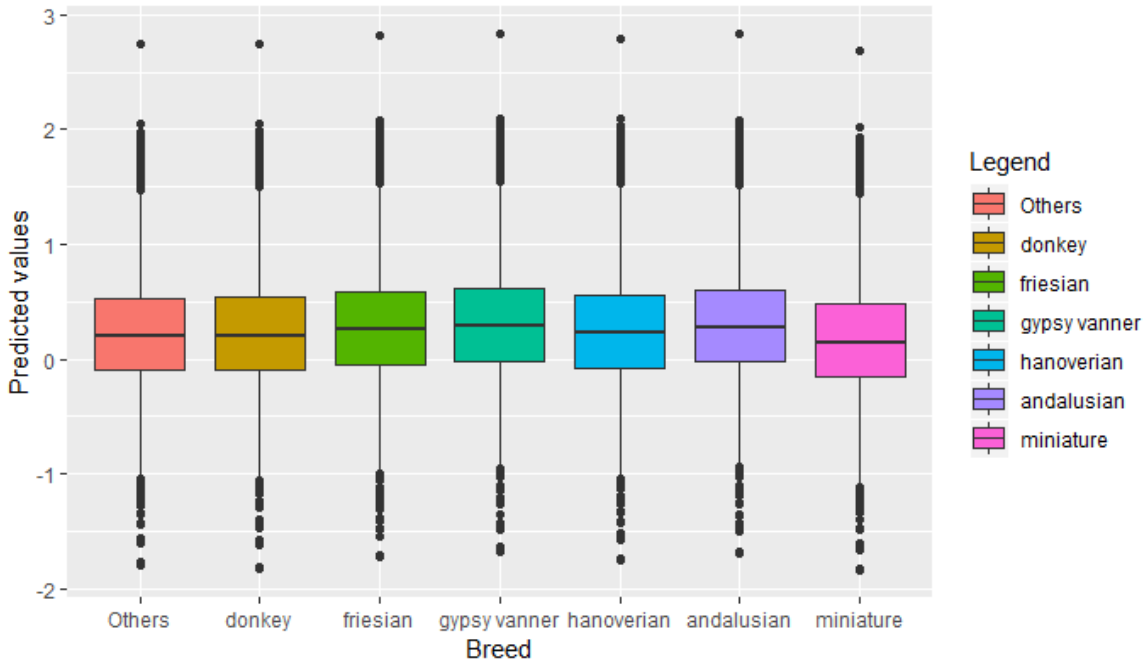

Figure 8: Explanation of breeds' effect over the end prediction.

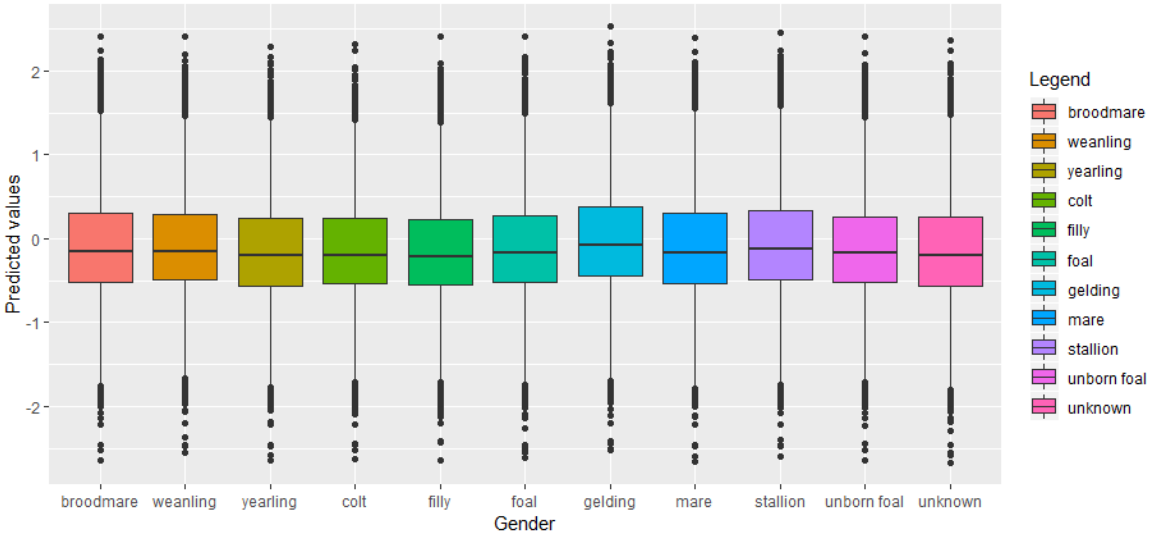

Figure 9: Explanation of genders' effect over the end prediction.

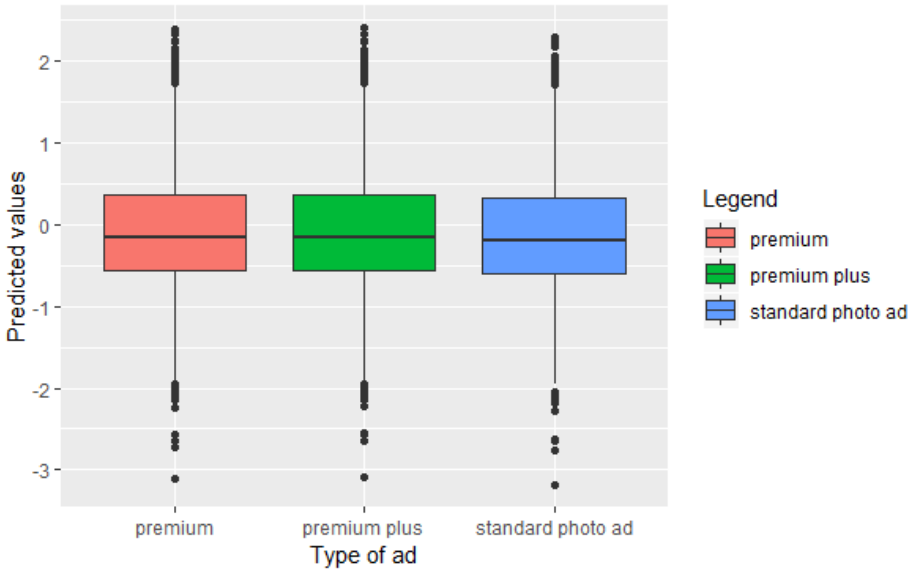

Figure 10: Explanation of the type-of-ad's effect over the end prediction.

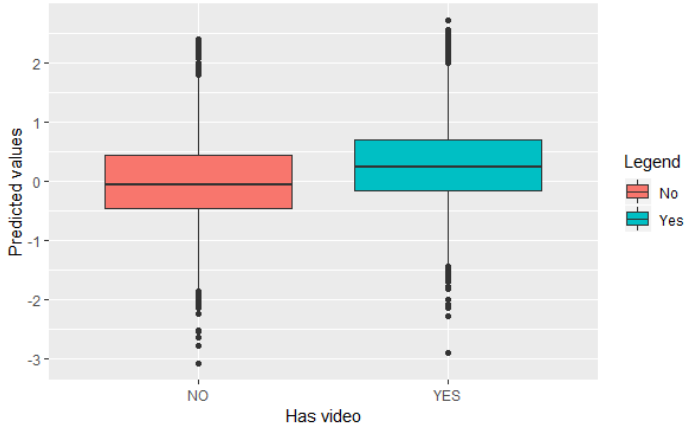

Figure 11: Explanation of the effect of having a video over the end prediction.

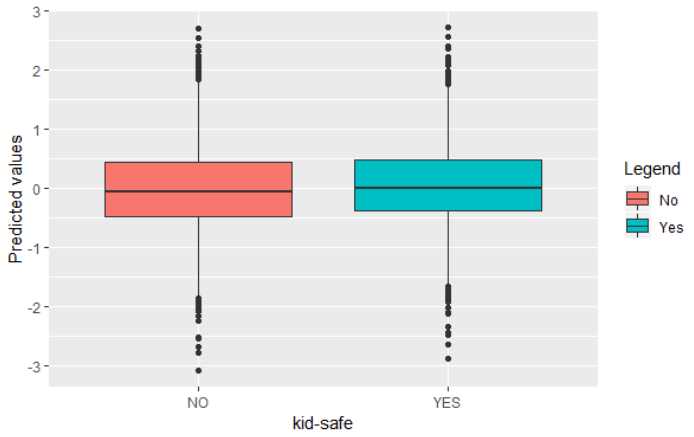

Figure 12: Explanation of the effect of being safe for children over the end prediction.

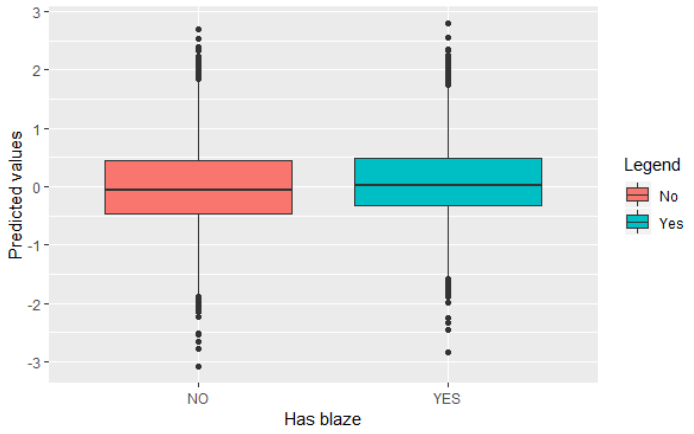

Figure 13: Explanation of the effect of having a blaze over the end prediction.

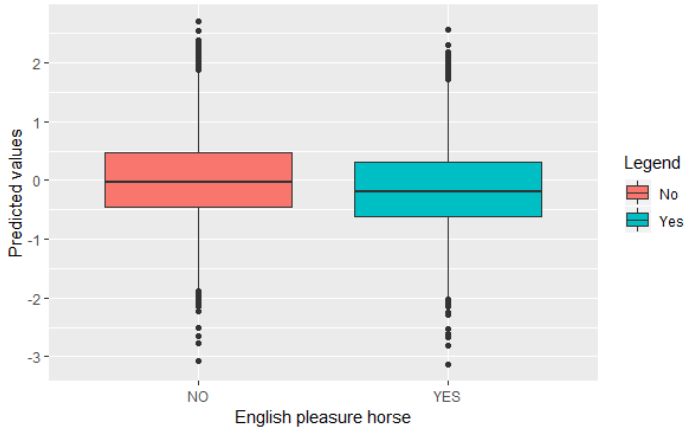

Figure 14: Explanation of the effect of being trained for English riding over the end prediction.

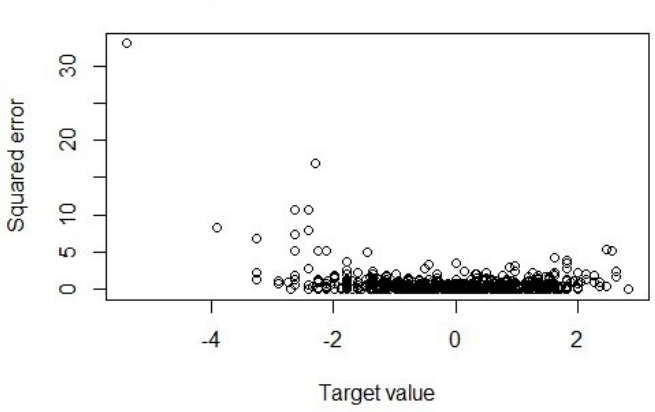

Figure 15: Distribution of the squared residuals of the text data unimodal model over the test set .

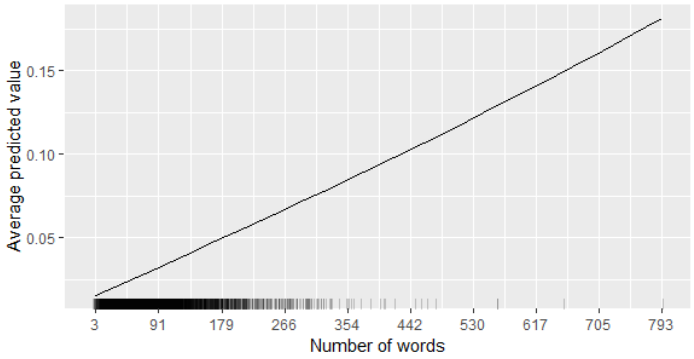

Figure 16: Partial dependence plot of the effect of the total number of words in an ad over the predicted price.

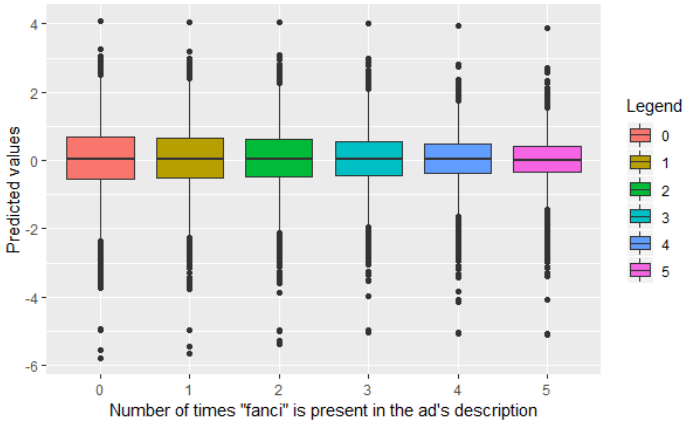

Figure 17: Explanation of the effect of the term "fanci" over the end prediction.

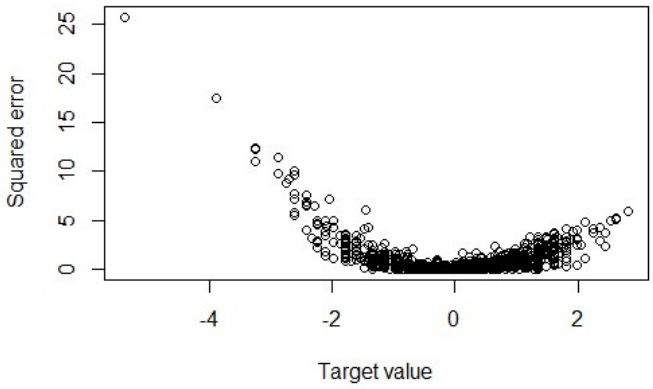

Figure 18: Distribution of the squared residuals of the image data unimodal model over the test set .

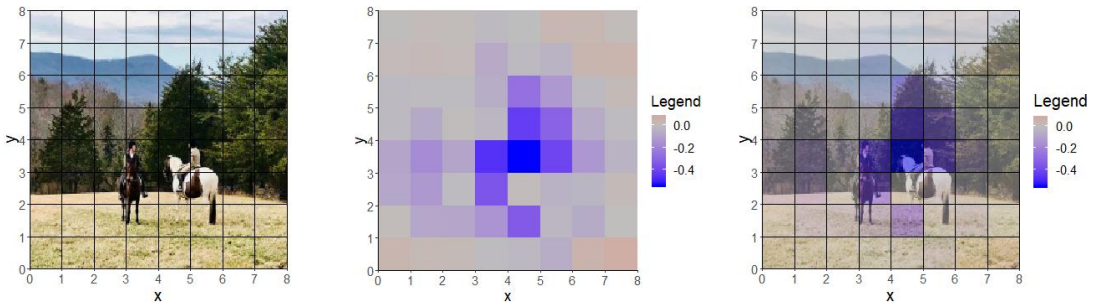

Figure 19: Sliding window heatmap for sample 1 of 20.

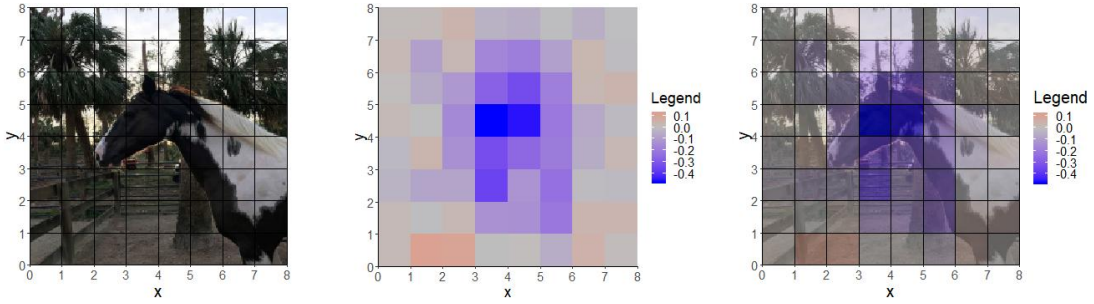

Figure 20: Sliding window heatmap for sample 2 of 20.

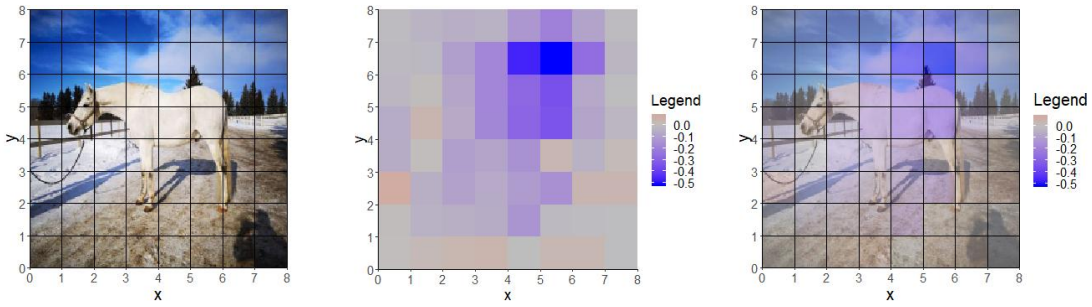

Figure 21: Sliding window heatmap for sample 3 of 20.

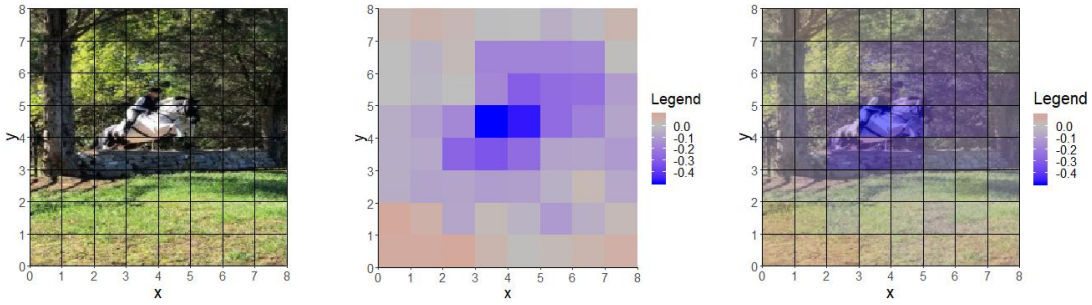

Figure 22: Sliding window heatmap for sample 4 of 20.

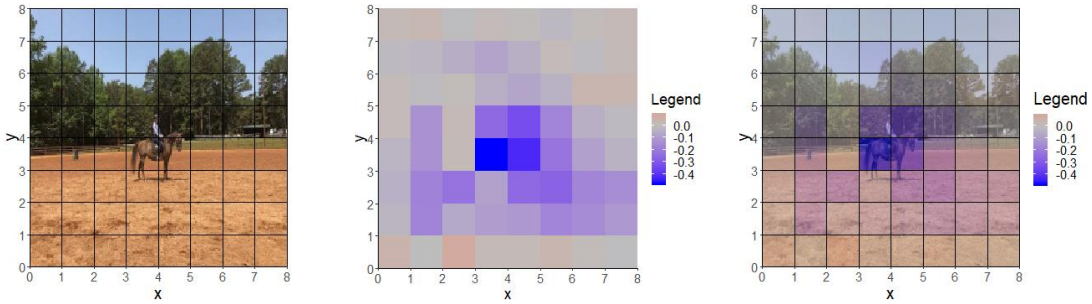

Figure 23: Sliding window heatmap for sample 5 of 20.

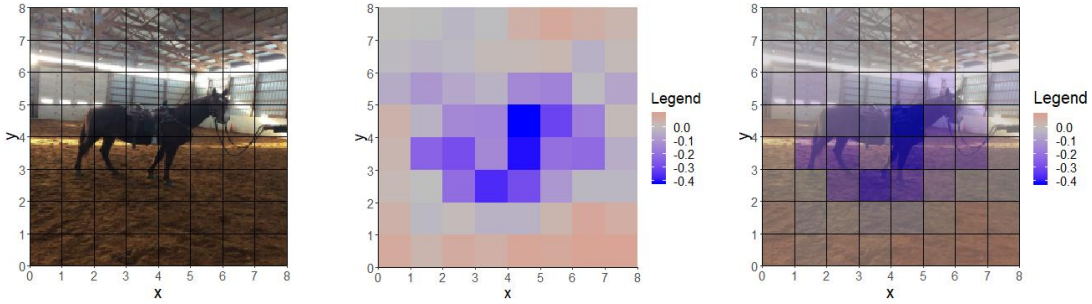

Figure 24: Sliding window heatmap for sample 6 of 20.

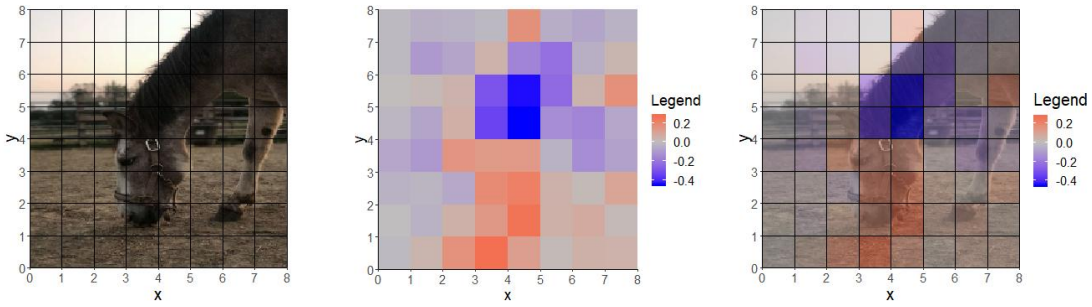

Figure 25: Sliding window heatmap for sample 7 of 20.

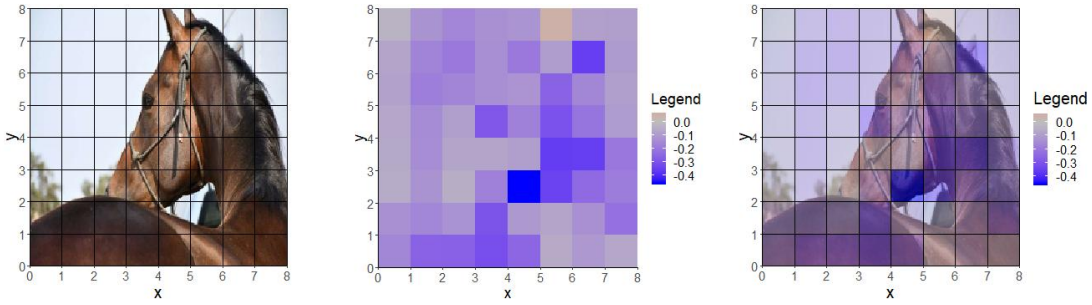

Figure 26: Sliding window heatmap for sample 8 of 20.

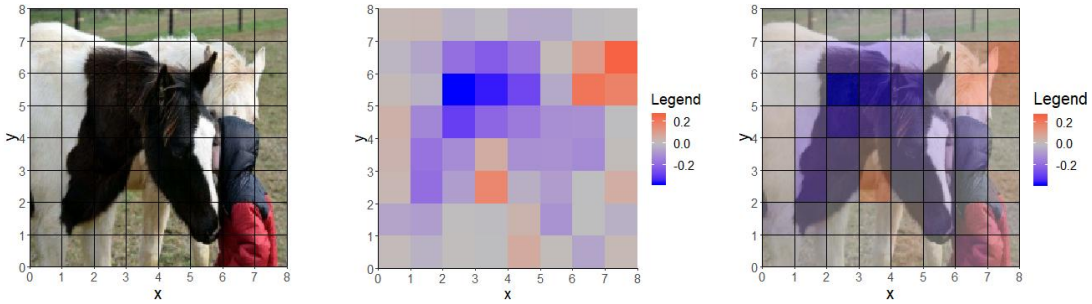

Figure 27: Sliding window heatmap for sample 9 of 20.

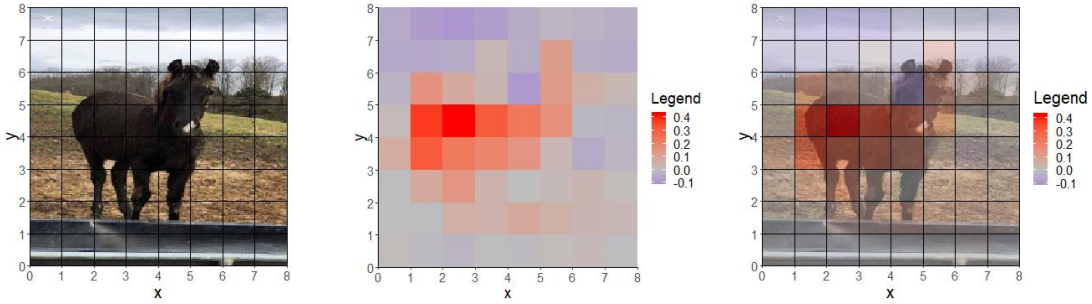

Figure 28: Sliding window heatmap for sample 10 of 20.

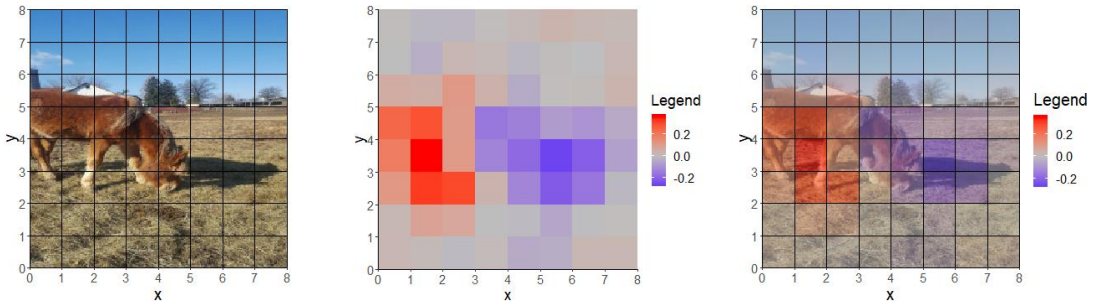

Figure 29: Sliding window heatmap for sample 11 of 20.

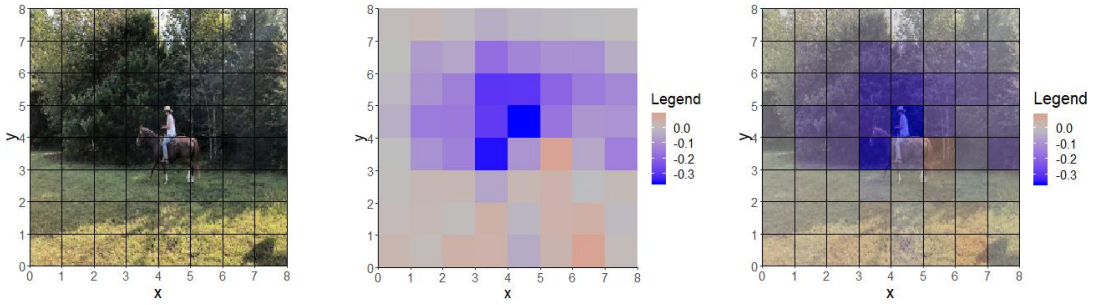

Figure 30: Sliding window heatmap for sample 12 of 20.

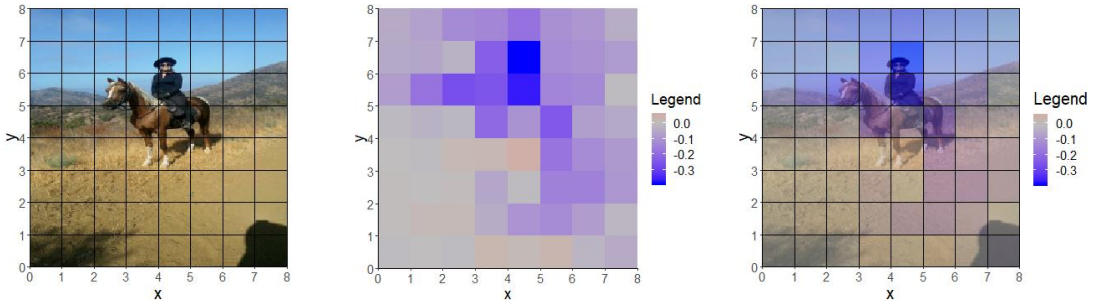

Figure 31: Sliding window heatmap for sample 13 of 20.

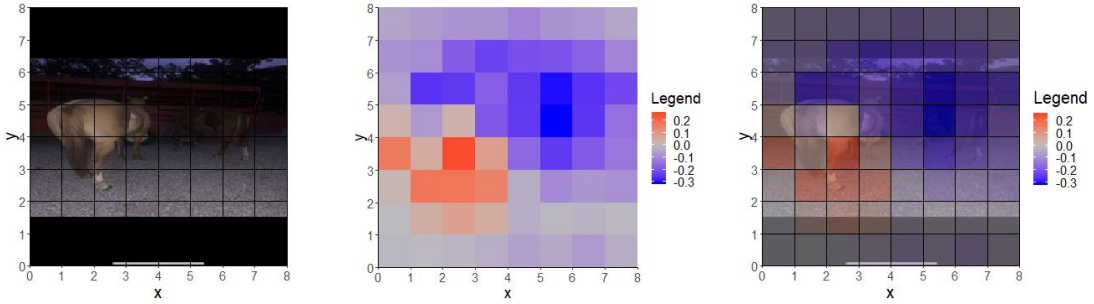

Figure 32: Sliding window heatmap for sample 14 of 20.

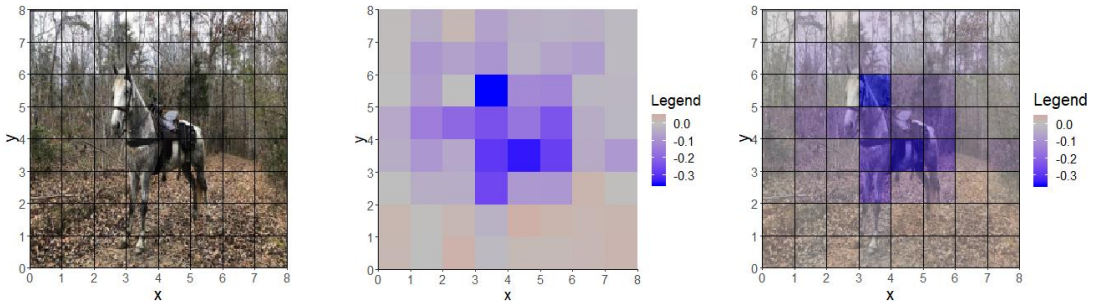

Figure 33: Sliding window heatmap for sample 15 of 20.

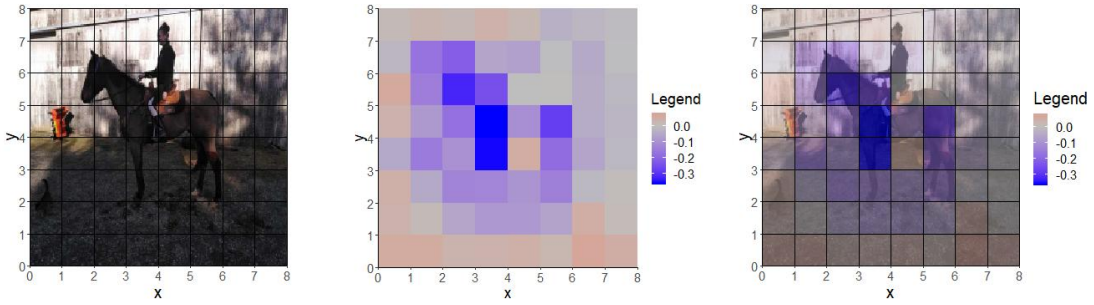

Figure 34: Sliding window heatmap for sample 16 of 20.

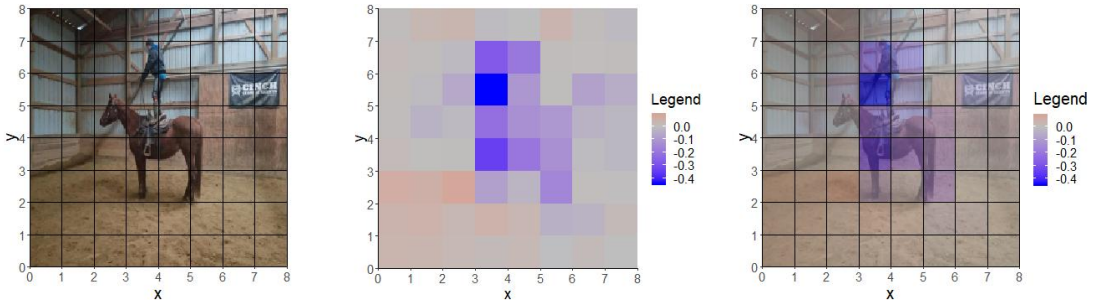

Figure 35: Sliding window heatmap for sample 17 of 20.

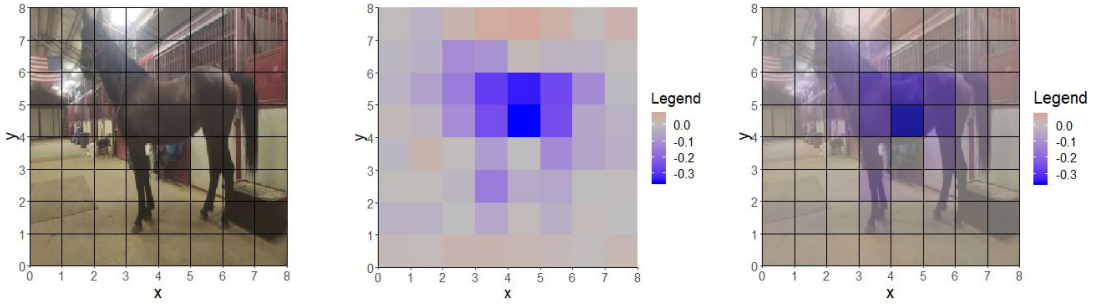

Figure 36: Sliding window heatmap for sample 18 of 20.

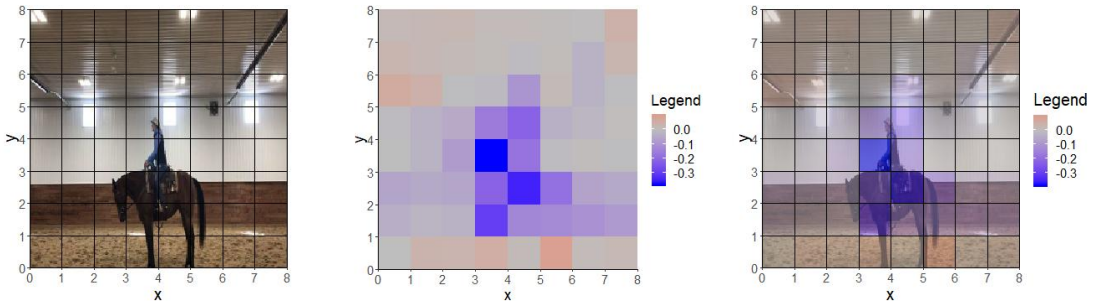

Figure 37: Sliding window heatmap for sample 19 of 20.

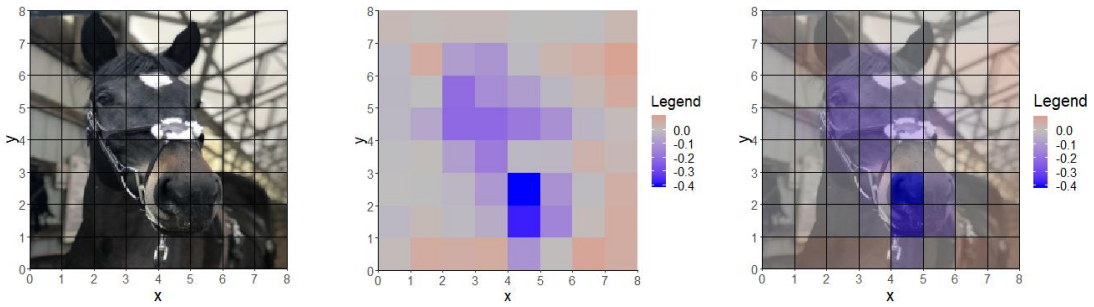

Figure 38: Sliding window heatmap for sample 20 of 20.

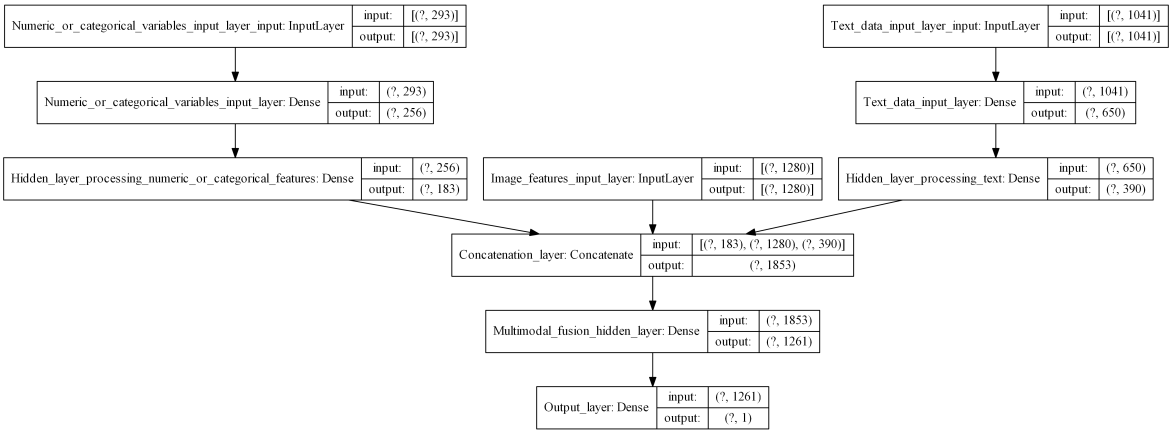

Figure 39: Visualization of the multimodal fusion optimal model's architecture.

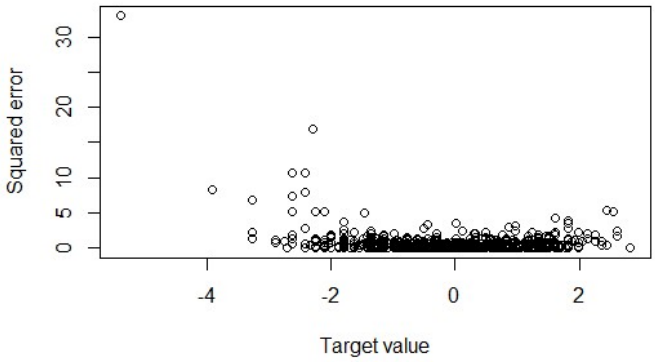

Figure 40: Distribution of the squared residuals of the multimodal model over the test set .

Appendix A

Rationale behind the choice of website.

The data required for the present study were not found in any database already available online. Thus, it was decided to web-scrape the data needed. This implied that a website from which to retrieve data had to be chosen. This is not an easy task. Indeed, there are multiple websites displaying advertisements for horses on sale and it is difficult to determine which one is best to use.

Freeborn (2009), who had to front the same problem in a previous research followed the sequent procedure. In 2008, she searched on Google the words "horses for sale". Then she proceeded to inspect the number of horses offered for sale on each of the first ten websites. Table 3 summarises the results of her web search. Finally, she decided to web-scrape the data from the website dreamhorse.com justifying this choice in three different ways. First, she states she has personally purchased and sold horses on different websites and the selected one had been the one offering the best services. Second, the website is among the ones displaying more advertisements, which is helpful when using statistical techniques which improve with the amount of data points. Third, and most importantly, the writer observes that *dreamhorse.com* is the only website in the top ten which allows to look for horses which have been sold on it in the past.

Table 5: Websites for Horse Trade in 2008 (Freeborn, 2009)

Rank	**Website**	**Number of horses**
1	Horsetopia	N/A
2	Equinenow	25,066
3	Horsefinders	4,950
4	Equine	N/A
5	Equinehits	69,840
6	Horseville	N/A
7	Horseclicks	15,460
8	Horsesforsale	N/A
9	Myhorseforsale	1,472
10	Dreamhorse	58,166

In the attempt to follow the same procedure, "horses for sale" was searched on Google on the 20th of March of 2020. Table 4 displays the result of the query. A quick comparison shows that the two tables are radically different. Only two websites appear on both (*Equine*, *Equinenow*). Moreover, the number of horses offered is far smaller in the website of the second table. This changes in the results might be due to the different location from which the queries were started (US the first, Netherlands the second), to a change in Google search algorithm or to the rise and fall of the different websites during the last twelve years. For instance, there are new websites, as *Dutchsporthorsesales* and *Winterhorses*, selling a limited number of high-quality horses at prices going from 44,000 to 445,000 euros.

Table 6: Websites for Horse Trade in 2020

Rank	Website	Number of horses
1	Equine	918
2	Ehorses	10,347
3	Dutchsporthorsesales	25
4	Equinenow	18,394
5	Horsezone	875
6	Winterhorses	23
7	Sporthorsenation.eve...	121
8	Ussporthorses	2,008
9	Germanhorsecenter	142
10	Horsedeals	283

By following the same procedure of Freeborn (2009), one of the websites in the top ten needs to be chosen. In the present paper one of the crucial criteria by which the websites should be assessed is the amount of data points that could be retrieved from it. Indeed, a large amount of data is needed to train computer vision algorithms. Moreover, even if it is not agreed on the minimum number of observations needed to train a convolutional neural network, it has been proven that its accuracy increases with the size of the training set (Lei, Zhang, Wang & Su, 2018). *Equinenow* and *Ehorses* are by far the two most promising websites under this point of view. To decide from which e-commerce platform retrieve the data a further exploration of the website content was done on the on the 26th March 2020.

It was found that *Equinenow* hosts ads regarding horses on sale in Canada or US. All the advertisements on the website are displayed for a time span of three months. At a preliminary examination, it seems that most of ads include some data describing the horse (i.e. height and breed), a picture and a lengthy text description in English. Finally, of the 18360 advertisements present of the website on the day of the exploration 740 were about stud fees which are not relevant for the present research, for 385 the price was not listed, 10573 were marked as sold. Unfortunately, when a horse is marked as sold the price is hidden on the website. Thus, since it is necessary to have the price of all the data entries to be able to build a hedonic pricing model, only 6662 ads of this website could be used in the present research.

Ehorses, instead, has got a more international focus, allowing to post listings to sellers from all over the world. An ad can be displayed on the website for a maximum of three months. For most of the horses a picture, some descriptive variables and a text description are given. However, the text is in general shorter than on the other website and it can be given in different languages, often in German. Finally, of the 10399 advertisements on the website on the 26th March 2020, for 665 the price was given only on application, for 170 the price was going to be sorted on an auction base, for 3612 only a price range was given, often a very large one, i.e. from €50,000 to €100,000. Eventually, only 5952 ads were found to display a single price. The sold horses are displayed in a dedicated page for 7 days after the sale and are then deleted from the website.

Eventually, *Equinenow* is found to be the most suitable website from which to retrieve data.

Indeed, even if choosing *Equinenow* over *Ehorses* limits the international scope of the research to only the United States and Canada, it also grants many advantages. First, it provides more data points (6662 vs. 5952). It is true that more data would be retrieved from *Ehorses* if, for the horses for which only a price range is available, a single price was obtained as the average of the maximum and minimum price. However, this would augment the data noise, especially since some of the price range are very large. A noisy input would lead to a noisy output; hence it is not convenient to adopt such a procedure. Second, the text description on *Equinenow* are longer, all in the same language and in English, all desirable characteristics when text analytics techniques are applied.

As an additional comment, it must be noted that of the 6662 ads present on *Equinenow* some might turn out not to have an image or a text description, hence the eventual number of useful data points could be lower. Moreover, if any more data will be eventually needed the website could be web-scraped multiple times at the distance of a couple of months given that apparently every three months almost twenty thousand new ads are uploaded, and ten thousand horses are sold.