



Article

Towards Virtual 3D Asset Price Prediction Based on Machine Learning

Jakob J. Korbel ^{*}, Umar H. Siddiq and Rüdiger Zarnekow

Information and Communication Management, Department of Economics and Management, Technische Universität Berlin, 10623 Berlin, Germany; uhsiddiq@googlemail.com (U.H.S.); ruediger.zarnekow@tu-berlin.de (R.Z.)

* Correspondence: jakob.j.korbel@tu-berlin.de; Tel.: +49-03-314-28571

Abstract: Although 3D models are today indispensable in various industries, the adequate pricing of 3D models traded on online platforms, i.e., virtual 3D assets, remains vague. This study identifies relevant price determinants of virtual 3D assets through the analysis of a dataset containing the characteristics of 135,384 3D models. Machine learning algorithms were applied to derive a virtual 3D asset price prediction tool based on the analysis results. The evaluation revealed that the random forest regression model is the most promising model to predict virtual 3D asset prices. Furthermore, the findings imply that the geometry and number of material files, as well as the quality of textures, are the most relevant price determinants, whereas animations and file formats play a minor role. However, the analysis also showed that the pricing behavior is still substantially influenced by the subjective assessment of virtual 3D asset creators.

Keywords: 3D model; virtual asset; virtual product; virtual good; pricing; machine learning; feature scoring; e-commerce; metaverse



Citation: Korbel, J.J.; Siddiq, U.H.; Zarnekow, R. Towards Virtual 3D Asset Price Prediction Based on Machine Learning. *J. Theor. Appl. Electron. Commer. Res.* **2022**, *17*, 924–948. <https://doi.org/10.3390/jtaer17030048>

Academic Editor:
Eduardo Álvarez-Miranda

Received: 24 May 2022

Accepted: 4 July 2022

Published: 7 July 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Digital 3D models are today indispensable in various industries. Manufacturers rely on 3D models to develop and simulate their products [1], retailers allow customers to configure product characteristics based on 3D visualizations [2], and game developers require 3D models not only to build their virtual worlds, but to gain profits through their purchase within these environments [3]. Technological trends such as augmented (AR) and virtual reality (VR) and ambitions from firms such as Meta to create a virtual metaverse [4,5] foster the importance of 3D models as vital building components, assets, and objects of trade. Consequently, marketplaces have emerged which focus on the trade of virtual 3D assets, i.e., 3D models that are not included in a virtual environment, and thus can be adapted for various fields of application [6]. Examples for virtual 3D asset platforms are the Unity Asset Store [7], which focuses on the trade of virtual 3D assets for the development of games and AR/VR environments; Thingiverse [8], which provides access to millions of 3D models to manufacture products based on 3D printing; and marketplaces such as CGTrader [9], Turbosquid [10], or Sketchfab [11], which offer virtual 3D assets for a variety of domains, e.g., e-commerce, architecture, or cultural heritage. However, whereas the pricing, value, and consumption of 3D models in virtual environments, i.e., virtual goods, has been extensively researched, studies considering the pricing determinants and value of virtual 3D assets are sparse, as are pricing recommendations in practice. Although some marketplaces provide basic pricing guidelines for 3D model creators, the information compromises general suggestions on 3D model characteristics to be considered rather than their specific relevance. In turn, creators and sellers of virtual 3D assets must set prices for their virtual 3D assets based on their subjective assessments.

Hence, the objective of this study was to identify relevant price determinants for virtual 3D assets and develop an IT artefact that considers these price determinants for

virtual 3D asset price predictions. Therefore, this study relied on the design science research (DSR) methodology, a dataset containing the meta-characteristics of 135,384 3D models from the Sketchfab marketplace (the largest platform for virtual 3D assets [11]), and a machine learning (ML) approach for the processing and analysis of 3D model characteristics.

To meet the research objectives, the paper is structured as follows. In the theoretical background section (Section 2), the characteristics of 3D models and their respective types are described, as well as the applied analysis approach, i.e., data mining and ML. Studies on virtual 3D assets are sparse; therefore, the related research section (Section 3) contains a summary of current approaches to identify and analyze price determinants based on ML in other disciplines, e.g., accommodation, cryptocurrencies, or the stock market, to derive an appropriate analysis framework for the study. The methodological approach is illustrated in Section 4, whereas the implementation of the artefact is described in Section 5. Finally, the results are evaluated and deployed in Section 6 and summarized in Section 7, concluding with future research avenues.

2. Theoretical Background

2.1. 3D Models and Virtual 3D Assets

The basis of 3D models is their geometry, produced out of meshes or bodies, representing the shape of an object which is often complemented with textures, materials, and, depending on the purpose, animations. The geometry represents the shape of a 3D model (Figure 1a); however, textures and materials are commonly used to create and modify the appearance of 3D models, to make the model more appealing or real, or to assign a meaning to an object (e.g., a 3D model in the shape of a sphere can become a volleyball by assigning the respective texture). Texture files (Figure 1b) are 2D images that “encompass” the 3D shapes, commonly based on UV mapping [12]. Materials, however, are more complex and include one or multiple texture files. In comparison with simple textures, materials allow the creator of a 3D model to adjust settings such as reflection, opacity, or bumps and wrinkles (Figure 1c) [13]. Hence, whereas textures represent the basic “skin” of a 3D model and require a material to be attached to the model, materials allow these skins to become more realistic. Today, the most advanced materials are physically based rendering (PBR) materials, because they facilitate the simulation of multiple material characteristics based on a complex mapping synthesis [14]. Apart from static characteristics, 3D models can be animated. Animations of simple objects can be realized by translating or rotating the object; however, more complex and 3D character animations, especially, require rigging [15]. Rigging refers to the process of including a “skeleton” in 3D objects that enables the animation of different parts within one 3D model (Figure 1d) [16]. Whether 3D models contain this information beyond their shape depends on the respective 3D file format. The STL format, for example, is the proprietary file format for 3D printing, and thus does not include texture, material, or animation information because those are not of use in a 3D printing process [17]. In contrast, all of this information can be stored in the FBX file format [18], a common format for media and game development. Hence, the characteristics of 3D models correlate with their anticipated usage and type.

The most common concepts that describe specific types of 3D models in industry are virtual products and virtual goods. Virtual products are 3D objects that represent actual physical goods in form (and function), and are especially useful in the manufacturing and retail domains. With the virtual representation of the actual product, different design variants of the product can be tested, virtual prototypes created, and real products simulated, at a fraction of the costs of physical processes [1,19,20]. In addition, virtual products have gained importance in the retail industry because 3D models allow for greater informativeness, enjoyment, or willingness to purchase if the 3D models are either used to allow the customers to configure their products based on their specific needs or to visualize the products in the real environment via AR to prevent, amongst others, incorrect purchase decisions [21,22]. In contrast, virtual goods represent usable and tradeable 3D models within virtual worlds and game environments [23,24]. Many game developers

today rely on the free-to-play business model; therefore, these virtual goods became one of the main revenue sources in the gaming industry [3], with an estimated turnover of about USD 190 billion in 2025 [25]. Due to their economic relevance, the value, pricing, and consumption mechanisms of virtual goods has been extensively researched. The results show that the value of virtual goods, and thus, their consumption, mainly derives from their characteristics within a closed virtual environment. Due to the closed environment, virtual goods can exhibit scarcity (albeit artificially created), the possession of multiple copies of the same virtual good can increase consumer utility, interactions with virtual goods can lead to higher purchase intentions, and the characteristics of the goods can create social distinctions [23,24,26–32]. Thus, the objective of virtual goods is trade and the generation of revenue through their purchase, while virtual products are a mean to an end to facilitate the creation and distribution of physical products.

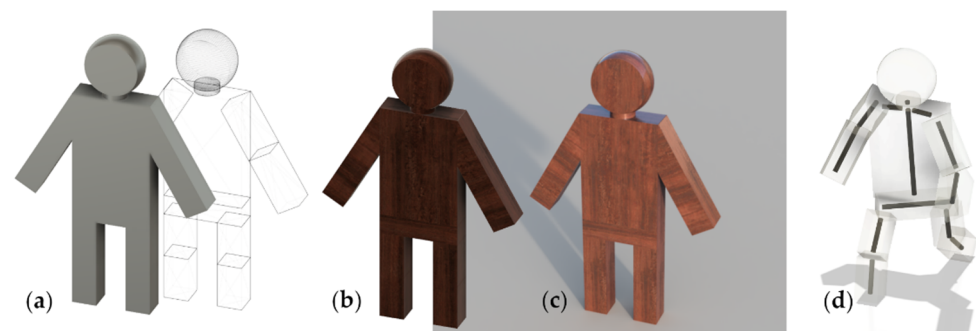


Figure 1. Three-dimensional model characteristics: (a) mesh/body; (b) body with texture; (c) body with texture and material settings (rendered); (d) body with rigged geometry for animation.

Virtual 3D assets, however, are both the antecedent of virtual products and goods and a necessity to create the environments in which virtual products and goods are used. In comparison with both concepts, virtual 3D assets neither have a predefined purpose nor are they already included in a virtual environment [6]. Hence, the value and pricing mechanisms of virtual goods cannot be transferred to virtual 3D assets because the virtual goods' value depends on its specific virtual environment. Virtual 3D assets are not traded in virtual environments, but on (real) online marketplaces, such as Sketchfab [11], to allow a wide range of customers, e.g., game developers, retailers, or architects, to build their virtual environments or use the virtual 3D asset as virtual goods, for example, by transferring and binding the 3D models to a specific environment. In contrast to communities that provide virtual 3D assets for free, e.g., Thingiverse (3D printing) [8], the offerings of virtual 3D asset marketplaces vary from simple, untextured low-poly models (for free), to complex 3D environments, such as city and building models for more than USD 5,000. In general, everyone can become a seller on the virtual 3D asset platforms by creating an account, considering the three largest marketplaces: CGTrader, Sketchfab, and Turbosquid. The marketplace providers receive a share of the selling price if virtual 3D assets are purchased. The pricing of the virtual 3D assets thereby remains with the seller. Although some marketplaces provide analytic tools to gather market insights (e.g., CGTrader [33]), the guidelines for sellers to set appropriate prices for their assets remain vague.

The guidelines (Table 1) suggest that the sellers should neither price their 3D models too high nor too low compared with similar 3D models so as to not undermine the store economy or imply a low quality of the virtual 3D asset. If higher prices are set, the description should allow the buyer to understand why the price is higher than for similar models. Furthermore, the number of file formats should play a role in the pricing process, as well as the quality of textures, or the optimization for games or AR/VR. According to the platform provider guidelines, textures should be in png format; the inclusion of editable file formats is advantageous, as are high-quality materials in form of PBR materials. In terms of animation, the seller should include as much animations as possible since models

with more animations sell better while the success heavily depend on the rigging of the model. However, the platform providers do not state which of the criteria are most relevant for potential customers and whether sellers actually consider these guidelines for their pricing decisions. Hence, an ML-based data mining approach was applied in this study to identify price determinants and allow the prediction virtual 3D asset prices.

Table 1. Pricing guidelines of the three dominant virtual 3D asset marketplaces.

Marketplace	Criteria	Pricing Guidelines
CGTrader [34]	Value for Buyers	"Consider the <i>value your work</i> brings to the <i>buyer</i> ".
	Price Range	"Make sure you <i>don't underprice</i> your model. Buyers might see it as a <i>sign of poor quality</i> ".
	Compatibility and Quality	"[. . .] make sure you provide a detailed description and preview images that showcase what distinguishes your model from the rest. That could be a <i>large selection of file formats, high quality textures, optimization for games or VR/AR, etc.</i> ".
Sketchfab [35]	Value for Buyers	"Consider the <i>value your work</i> offers a <i>potential customer</i> ".
	Price Range	"You can browse the store for <i>similar models</i> to guide your <i>pricing decision</i> ". "Be careful <i>not</i> to radically <i>undercut</i> the <i>price of similar models</i> on the store. This ultimately hurts all sellers by undermining the store economy". " <i>Low pricing</i> can be interpreted by buyers as a <i>sign of poor quality</i> ". "Similarly, be aware that <i>asking for significantly more</i> than similar models from other contributors can lead to <i>reduced sales</i> ".
	Compatibility and Quality	"If you set a <i>higher price</i> than similar models from other sellers, use the model <i>description</i> to explain <i>what distinguishes your model</i> and adds to its value. For example, the inclusion of <i>higher resolution textures</i> or <i>multiple file formats</i> would be an added benefit".
	Compatibility	"The <i>more file formats</i> you include, the <i>more successful</i> you will be". "The <i>ideal texture format</i> for textures is PNG. Buyers will also appreciate the inclusion of <i>Photoshop, Gimp, or similar editable layered files</i> ".
	Quality	"A complete set of PBR textures (Albedo, Metallic, Roughness) and normal maps are <i>desirable</i> to buyers for contemporary game engines".
	Animation	"Models with more animation states sell better". "The success of <i>animated models</i> is often very <i>rig dependent</i> . Be sure to use our additional files feature to include <i>rigged versions</i> in popular <i>software formats</i> ".
Turbosquid [36]	Price Range	"Setting your prices <i>extremely low</i> will <i>not</i> necessarily lead to <i>better sales</i> ". "Make sure you are pricing your models to achieve maximum sales and royalties. Look at <i>comparable 3D models</i> on the site to check their prices".
	Compatibility, Quality and Animation	"Realism" "File formats offered" "Texture/material/rigging settings"
	Complexity	"Complexity" "Poly count"

2.2. Data Mining and Machine Learning

Data mining describes the process of identifying correlations and patterns in data to create insights that add to existing knowledge [37]. It is a well-established method in e-commerce research to derive analysis results regarding, for example, product pricing, by acquiring and examining datasets from online marketplaces and platforms; therefore, this is an eligible approach for the explorative objectives of this study. To allow the extraction of knowledge from a given set of raw data, data mining comprises (1) data preparation, (2), data pre-processing, (3) data analysis, and (4) data interpretation [37,38].

The (1) data preparation phase focus on the identification of the most appropriate data sample for the data analysis process [39]. In this process, the nature of data is analyzed, and the appropriate data sample is thereafter extracted from the context of examination, e.g., online websites [40].

(2) Data pre-processing refers to all processes related to the cleaning, transformation, and selection of eligible data for the data mining task [38,41]. First, data cleaning is a vital step in data pre-processing because real world data can be unstructured and of a heterogeneous and noisy nature. Incomplete or inaccurate data due to missing values or outliers, for example, significantly impact the results of any subsequent steps in the data analysis [42]. Hence, inaccurate data must be treated by filling missing data with global constants or erasing outliers and gross errors [38,39]. Second, data transformation is required because variables in the dataset can contain different data types and ranges that affect the analysis process. In the data transformation step, data can be normalized to adjust the data to a common data range or smoothed by discarding data based on min/max ranges [38]. Third, a dataset can include redundant variables or variables which are irrelevant for the prediction of the target variable. These variables can impact the analysis process negatively because they do not provide additional useful information, adding bias to the data, increase the dimensionality, or significantly decrease the prediction performance of the analysis models for “unseen” data, i.e., overfitting [43–45]. Therefore, irrelevant variables must be detected and excluded from the data sample.

Finally, the dataset is (3) analyzed and (4) interpreted. ML has emerged as a promising alternative to common data analysis methods, especially when working with large data samples. ML-based data mining can be divided into (a) the data preparation phase, i.e., data transformation, exploration, and feature engineering, (b) the modelling phase, i.e., training and test cycles, (c) the evaluation phase, i.e., performance measuring and model selection for deployment, and (d) the deployment phase, i.e., the usage of the trained ML model [46]. The (a) data preparation phase corresponds with the previously described generic data pre-processing phases in data mining. The (b) modelling phase comprises both the selection of appropriate ML algorithms as well as their training and performance optimization [39]. The applied ML models range from supervised to unsupervised and reinforcement learning [47]. It is difficult to predict the best performing ML algorithm for a given dataset beforehand; therefore, it is common practice to apply and evaluate different ML algorithms with varying parameters to identify an eligible model for the ML process [46]. To (c) evaluate the model performance, several methods have been developed and applied to ML models. Amongst others, a common approach for the evaluation process is k-fold cross-validation, in combination with error metrics such as mean absolute error (MAE), the mean squared error (MSE), root-mean squared error (RMSE), coefficient of determination (R^2), or adjusted coefficient of determination (aR^2). K-fold cross-validation is used to validate the performance of ML models in terms of their ability to predict new, unseen data by splitting the training dataset into k subsets and applying the model k times while selecting a new validation set in every iteration [48]. After every iteration, error metrics or performance measures are calculated, which are used to evaluate the prediction performance of a model. Finally, the ML process is completed by its (d) deployment through the selection and exploitation of the best performing ML model [46], and conclusions for the specific field of application can be drawn by interpreting the resulting data.

3. Related Studies

The literature on pricing and the identification of price determinants for virtual 3D assets is sparse. However, previous work emphasizes approaches to examine price determinants based on ML for other fields of application, ranging from accommodation [49–51] and the stock market [52,53], to e-commerce [54,55], cryptocurrencies [56–58], and energy prices [59]. Apart from dynamic pricing approaches based on reinforcement learning [60,61], most publications have focused on the application of supervised learning algorithms to identify price determinants. The procedures for the data analysis vary between the super-

vised learning approaches, depending on the given dataset and the applied ML models (Appendix A, Table A1).

First, the datasets are described and pre-processed by removing irrelevant features and missing values and transforming the data into an appropriate format for the ML process, such as by converting the data into a consistent file format or normalizing skewed data based on log-transformation [49,52]. Second, preliminary analyses are applied to the dataset to enable clustering of the data and the first insights into correlations between the predictors and the target variable. Third, ML models are trained and tuned with the resulting dataset. Therefore, authors rely on a variety of machine learning models, ranging from tree-based approaches to support vector regression (SVR) and neuronal networks, whereas linear regressions are often used as baselines for subsequent evaluations (Appendix A, Table A1). Additionally, the models are evaluated by performance measures and feature importance scores that enable ranking of the identified price determinants. For the evaluation, most publications rely on k-fold cross-validation in combination with common error metrics, e.g., (R)MSE, MAE, and R^2 . To gain insights into the most relevant price determinants, feature importance, i.e., variable importance, methods can be applied to rank the variables according to their relevance in the ML modelling process [50,52,53,62]. The findings in the studies provide evidence that the approaches lead to promising results for both identifying price determinants and predicting prices in the respective field of application. Hence, the approaches in previous publications were adapted in this study for the identification of relevant price determinants and the implementation of the virtual 3D asset price prediction tool based on a DSR approach.

4. Methodology

This study followed the guidelines of the design science research (DSR) methodology based on Hevner et al. [63] and Peffers et al. [64,65], relying on an explorative data analysis based on the data mining approach. The DSR methodology is a problem-solving process which allows researchers to acquire knowledge and understanding of a problem domain and its solution through the creation and application of IT artefacts [63]. Peffers et al. [64,65] developed a conceptual process for DSR in information systems based on theoretical frameworks in design research studies. The design science research process (DSRP) consists of six distinct activities, which are performed in sequential order, while detailing the expected output of each step (Table 2). The DSRP is closely related to the seven DSR guidelines of Hevner et al. [63], because it also stresses the importance of an adequate and suitable problem statement, the construction of a viable solution, and the evaluation of results regarding their usefulness as well as their communication.

Table 2. Design science research process (DSRP) based on Peffers et al. [64,65] and a description of associated work in this study.

#	Process Step	Description
1	Problem identification and motivation	Three-dimensional models are widely used and gaining more significance through current technology trends; however, pricing determinants for virtual 3D assets are unknown and price predictions are therefore not feasible.
2	Objective of a solution	Identification of relevant price determinants and development of a price prediction model for virtual 3D assets.
3	Design and development	Dataset containing 135.384 3D model characteristics, univariate and bivariate analysis, feature engineering, and ML modelling.
4	Demonstration	Demonstration on the case of Sketchfab.
5	Evaluation	Evaluation of price determinants based on bivariate analysis, evaluation of ML models based on MAE, MSE, RMSE, R^2 , and aR^2 .
6	Communication	Front-end application for virtual 3D asset price prediction.

DSR requires a fundamental understanding of the research problem and the capability to find potential solutions towards solving the problem to provide a valuable contribution. The (1) problem identification and motivation as well as the (2) objectives of the solution, i.e., the creation of an artefact for the identification of price determinants and the prediction of virtual 3D asset pricing, are described in Sections 1–3. The (3) research design and development is based on the considerations of related literature (Section 3) and constitutes the technical framework for the implementation of the artefact. The implementation process was divided into four subsequent steps premised on the guidelines for ML-based data mining: (a) data preparation, (b) data pre-processing, (c) ML modelling, testing, and evaluation, and (d) prediction tool development. In the (a) preparation phase, a dataset including the characteristics of 135,384 3D models was crawled from the website Sketchfab based on the open-source web-crawling framework Scrapy [66]. As discussed in the theoretical background section, there are multiple online platforms trading virtual 3D assets. However, very few marketplaces focus on virtual 3D assets in general without an anticipated field of application and offer relevant metadata in the form of 3D model characteristics. Thereby, the virtual 3D asset marketplace Sketchfab offers the most comprehensive set of metadata, which can be utilized to predict 3D model prices. In addition, Sketchfab is the largest platform for virtual 3D assets with more than three million 3D models published and around three million registered users [11]. Subsequently, the (b) data are pre-processed by data cleaning and transformation in the form of feature engineering to identify potential price determinants using the libraries Pandas [67], Numpy [68], Matplotlib [69], and Seaborn [70]. The pre-processed data, including 118,358 3D models, provide the basis for the (c) ML modelling and tuning process. For this process, we relied on the libraries SciKit-Learn [71] and XGBOOST [72], as well as ML algorithms that are commonly used for price determination and prediction (Appendix A, Table A1): regularized linear regression (Lasso and Ridge regression), decision tree, random forest, extreme gradient boosting trees, and support vector regression (Appendix B, Table A2).

To (4 and 5) demonstrate and evaluate the artefact, the ML models for the price prediction task were trained, tuned, and evaluated based on the Sketchfab dataset. For the evaluation, the common error and performance metrics MAE, MSE, RMSE, R^2 , and aR^2 were applied (Appendix B, Table A2). In addition, we relied on feature importance scoring by embedded methods [43–45] and the mean decrease impurity (MDI) metric to identify relevant price determinants. Lastly, the results were (6) communicated in form of a (d) front-end application, based on the selection and application of the ML model with the best performance.

5. Results

Aggregation and analysis of the data were conducted in three phases: the (Section 5.1) data preparation phase, i.e., data extraction and selection, the (Section 5.2) data pre-processing, i.e., data cleaning, exploratory data analysis, and feature selection and engineering, and (Section 5.3) ML model training and performance optimization.

5.1. Data Preparation

Today, more than three million virtual 3D assets are available on Sketchfab, and although most models are view only, over 500,000 models are downloadable for free, with only a fraction available for purchase in the Sketchfab store [11]. This study focused on price determinants and prediction, only the data of virtual 3D assets that refer to the latter category were obtained. In addition, only data were considered that were relevant for the research objective, i.e., characteristics regarding the geometry, appearance, animation, or the compatibility of the 3D models. Hence, data such as customer reviews, categories, or tags were neglected.

To obtain the data, the developer tools of the Firefox10 web browser were used to analyze the 3D model listing and cursor referencing. Sketchfab lists up to 24 models on each page (per default), whereas every page is loaded by an HTTP GET request to the host-

server API, which includes the URL of the store along with the “sort-by” parameter and the current (non-numeric) cursor value. The host server responds with a JSON response, including up to 24 listings of the 3D models referenced under the current cursor value. Furthermore, the JSON response contains a reference to the previous and next cursor values, as well as the URLs of the previous and next pages. If the current page is the first or the last page in the pagination, the reference to the next and previous cursor values, as well as pages, is set to the null value. Hence, this URL can be used for parsing through the pagination logic, until the last page is reached, and crawling all “uid” (unique ID) values of the 3D models listed on the Sketchfab store. The JSON response contains all relevant metadata of a particular 3D model and resulted in 135,384 entries representing the characteristics of the 3D models. Although there is no information on the website about the actual amount of purchasable virtual 3D assets, the described crawling process should enable the extraction of all available 3D models on sale. In addition, the number of crawled items deems adequate for the derivation of price determinants and prediction. The initial dataset consists of 21 columns representing the features of the virtual 3D assets (Table 3).

Table 3. Virtual 3D asset characteristics data (Sketchfab store).

Feature	Data Type	Category	Description
model_price	float64	Currency	Price of the 3D model in the Sketchfab store
3d_model_size	float64	Geometry	File size of the 3D model in Sketchfab format
face_count	float64	Geometry	Number of faces in 3D model
lines	float64	Geometry	Number of lines in 3D model
morph_geometries	float64	Geometry	Number of morph geometries in 3D model
polygons_count	float64	Geometry	Number of polygons in 3D model
points	float64	Geometry	Number of points in 3D model
quads_count	float64	Geometry	Number of quads in 3D model
total_triangles_count	float64	Geometry	Number of total triangles in 3D model
triangles_count	float64	Geometry	Number of triangles in 3D model
vertices_count	float64	Geometry	Number of vertices in 3D model
materials_count	int64	Appearance	Number of materials attached to the 3D model
pbr_type	String	Appearance	Physical based rendering characteristics (material)
textures_count	int64	Appearance	Number of textures attached to the 3D model
total_texture_sizes	int64	Appearance	File size of textures attached to 3D model
uv_layers	Boolean	Appearance	UV layers for texturing in 3D model
vertex_color	Boolean	Appearance	Vertex colors in 3D Model
animations_count	int64	Animation	Number of animations attached to the 3D model
rigged_geometry	Boolean	Animation	Rig or “skeleton” of the 3D Model for animation
scale_transformation	Boolean	Configuration	Allows scale configuration of 3D model
file_format	String (Array)	Compatibility	3D model file format

5.2. Data Pre-Processing

The data pre-processing was conducted in three steps. First, the data were pre-processed for the feature selection. Thereby, the features of the dataset were analyzed individually based on a univariate data analysis to examine anomalies and outliers. In a subsequent step, bivariate data analysis was conducted to identify interdependencies between the target variable, i.e., the model price, and potential numeric predictor variables, as well as the relationship between the predictor variables. Second, relevant features were selected based on the analysis results. Third, the features were engineered based on feature encoding, feature transformation, and scaling methods.

Before the analysis, the features were examined with regard to their missing values. All features showed missing values. The feature *pbr_type* had the highest frequency of missing values (88,192) due to the non-existent classification requirement of Sketchfab. Three-dimensional models can have one of two PBR types (metalness and specular), or no PBR type. Hence, all missing values in the category *pbr_type* were replaced with the string

value “None”. The missing values in the remaining 20 features varied from 16 to 21 and 54, apart from the feature *file_format* (4.485). The remaining number of missing values was low compared with the total size of the dataset; therefore, we assumed that their impact on the data sample was insignificant [73]. Thus, the rows containing the missing values were removed. After treating the null values, 130.862 samples remained in the dataset.

5.2.1. Univariate and Bivariate Data Analysis

In a first step, the features in the dataset were individually analyzed to gain an understanding of the value distributions based on a univariate data analysis. Examining the distribution of variables’ values is mandatory to adequately explore the data and gained insights about possible anomalies and outliers that can negatively impact the learning process [74].

In the numeric features (Table 4), it is apparent that the distribution of the target variable *model_price* has a significant positive skewness, while most models have a price beneath USD 1.000. A minimum price of USD 3,99 is determined by Sketchfab [35]. In addition, the distribution has a leptokurtic kurtosis. This is not unexpected, because 75% of the 3D model prices in the dataset are beneath USD 20, although the values range up to a maximum of USD 5.500. In terms of the variables that describe the geometry of the 3D models, the analysis showed that the minimum value for *total_triangle_count* is 0, which implies that the 3D model does not contain a constructed 3D geometry. An examination of the entries with *total_triangle_count* = 0 (127) revealed that the corresponding models consisted of point clouds from 3D scans with *points* > 0. Apart from these 127 entries, only 6 models remained with *points* > 0. As for the distribution of *model_price*, the values of *total_triangles_count* were significantly skewed right and leptokurtic. Similar significantly leptokurtic and positively skewed distributions of values can be observed for all numeric variables. As an example, the distribution of *model_price* is illustrated in Figure 2. In the appearance-related variables, 2.160 3D models have a *total_texture_size* > 0, whereas their *textures_count* is 0, which can be considered an anomaly because textures can only have a size if texture files exist. Furthermore, in terms of explanatory value, the variable *total_texture_size* is less conclusive because it expresses the aggregated size of the texture files rather than providing information about the data size of single texture files, and thereby, their quality. Therefore, we include the variable *textures_mean_size*, which depicts the mean texture sizes in every 3D model (*total_texture_size/textures_count*).

Table 4. Numeric features and values.

#	Numeric Feature	Mean	Std	Min	Max	25%	50%	75%
F1	model_price	20,06	39,03	3,99	5.500,00	5,00	10,00	20,00
F2	3d_model_size	2.168,80	5.632,85	0,06	201.928,43	60,07	268,84	1.510,93
F3	face_count	261.523,60	637.181,00	0,00	24.346.160,00	7.000,00	35.062,00	200.541,00
F4	lines	78,68	4.024,83	0,00	565.115,00	0,00	0,00	0,00
F5	morph_geometries	0,03	1,09	0,00	228,00	0,00	0,00	0,00
F6	polygons_count	165,59	2.016,96	0,00	164.428,00	0,00	0,00	0,00
F7	points	2.258,89	102.193,90	0,00	18.680.440,00	0,00	0,00	0,00
F8	quads_count	59.432,71	225.627,90	0,00	9.387.565,00	0,00	873,00	16.585,00
F9	total_triangles_count	261.523,20	637.183,70	0,00	24.346.160,00	7.000,00	35.056,00	200.541,00
F10	triangles_count	141.452,10	464.422,70	0,00	24.346.160,00	116,00	3.384,00	50.000,00
F11	vertices_count	142.457,40	357.146,80	0,00	12.529.260,00	3.805,00	18.804,00	108.254,00
F12	materials_count	5,07	9,02	1,00	100,00	1,00	2,00	5,00
F13	textures_count	6,48	11,97	0,00	443,00	1,00	4,00	6,00
F14	textures_mean_size	3.898,17	7.611,02	0,00	203.399,44	168,90	1.292,63	4.262,19
F15	total_texture_sizes	21.058,44	48.438,11	0,00	1.809.410,00	498,26	6.111,65	21.887,88
F16	animations_count	0,34	3,80	0,00	312,00	0,00	0,00	0,00

In the categorical features (Table 5), the analysis of the distribution revealed that over one-third of the 3D models had a PBR type (32,48% “metalness”, 3,25% “specular”), 92,86% had UV layers, 13,65% had vertex colors, 6,19% had a rigged geometry, and only 4,48% had an option for scale transformation. To gather insights into the compatibility of 3D models, the variable *file_format* was transformed before analysis of the variable *file_format_score* that

represents the number of file formats that a single entry contains rather than the denotation of every file format in the entry. Of the 3D models, 75,47% are only available in a single file format, 15,02% have two file formats, 4,38% have five file formats, 2,64% have four file formats, and 1,13% have three file formats. The remaining entries (1,36%) provide their models in 6 to 15 file formats.

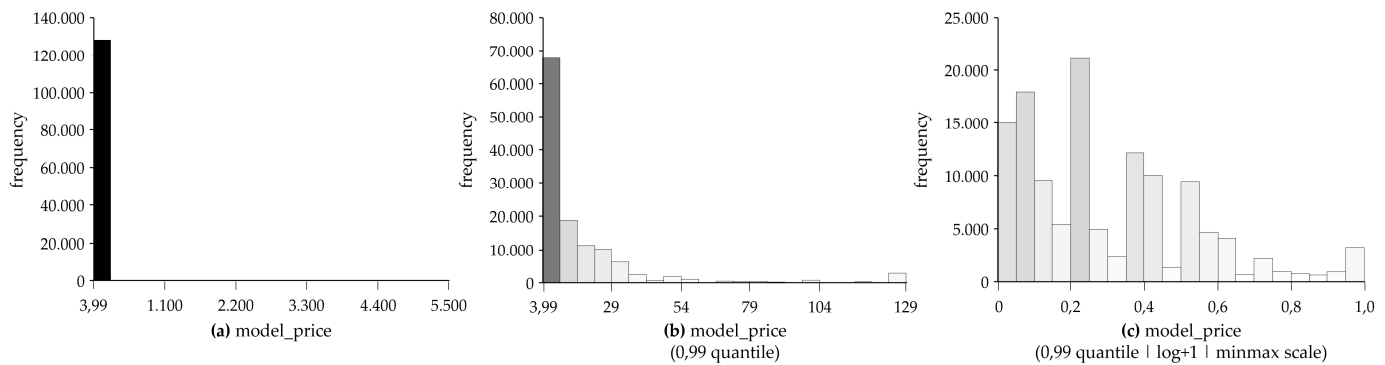


Figure 2. Histograms of *model_price* value distributions: (a) *model_price* value distribution; (b) *model_price* value distribution after 0,99 quantile criterion; (c) *model_price* value distribution after logarithmic transformation and normalization.

Table 5. Categorical features and values.

#	Categorical Feature	Values
C1	pbr_type	none (84.099), metalness (42.506), specular (4.257)
C2	uv_layers	true (121.518), false (9.344)
C3	vertex_color	true (17.867), false (113.009)
C4	rigged_geometry	true (8.096), false (122.766)
C5	scale_transformation	true (5.862), false (125.000)
C6	file_format_score	1 file format (98.766), 2 file formats (19.649), 3 file formats (1.484), 4 file formats (3.452), 5 file formats (5.731), 6 file formats (1.307), 7 file formats (378), 8 file formats (26), 9 file formats (30), 10 file formats (22), 11 file formats (7), 12 file formats (3), 13 file formats (4), 14 file formats (2), and 15 file formats (1)

Based on the insights, anomalies and outliers could be removed from the dataset. First, the 127 samples with *total_triangles_count* = 0 were dropped from the dataset, as was the variable *points* and the remaining 6 entries with *points* > 0. Furthermore, the entries that had a *textures_count* = 0 but a *total_texture_size* > 0 were excluded from the dataset as anomalies (after the removal of the 133 point-related entries: 2.155). Second, the target variable *model_price* was significantly skewed and with a leptokurtic kurtosis, as were the distributions of all numeric features. Hence, to enable consistent prediction performance, only values equal to or smaller than the 0.99 quantiles of these features were considered for the subsequent analysis (for example, *model_price*: Figure 2). The 0.99 quantiles of *morph_geometries* and *lines* are 0; therefore, these variables were excluded from the feature set. The resulting dataset consisted of 118.358 entries.

In a second step, the relationships between the target variable, i.e., the model price, and the remaining numeric and categorical predictor variables, as well as the relationships between the numeric variables and the categorical variables, were examined based on a bivariate data analysis. Most numeric variables in the dataset were neither normally distributed nor linear; therefore, a non-parametric correlation coefficient, the Kendall's rank correlation coefficient (enhancement of the Spearman's rank correlation coefficient), was chosen for the analysis of the numeric values. The coefficient measures the statistical dependence between

the ranked values of variables instead of the raw data. Therefore, correlations between non-linear variables could be described, as long as their relationship was monotonic in nature (or followed a monotonic function) [75,76]. In addition, Kendall's rank correlation coefficient measured the strength of the ordinal association between two variables by calculating a normalized score for the number of matching (concordant) rankings [75]. The correlation matrix is illustrated in Table 6. The highest correlation values between the target and numeric variables could be observed in the geometry-related features *3D_model_size* (0.27), *face_counts* (0.25), *vertices_count* (0.25), and *total_triangles_count* (0.25). In addition, the results showed that *total_triangles_count* had a perfect correlation with *face_count* (1.00) and significant positive correlations with *vertices_count* (0.97), *3D_model_size* (0.86), *triangles_count* (0.43), and *quads_count* (0.26). In the appearance-related features, *material_count* was positively correlated with *textures_count* (0.17) and showed negative correlations with *textures_mean_size* (−0.22) and *total_texture_size* (−0.08). The *total_texture_size*, however, was highly correlated with the *textures_count* (0.49) and *textures_mean_size* (0.74), whereas the *textures_mean_size* was positively correlated with *textures_count* (0.23).

Table 6. Kendall correlation matrix (numeric features).

#	F1	F2	F3	F6	F8	F9	F10	F11	F12	F13	F14	F15	F16
F1	1.00												
F2	0.27	1.00											
F3	0.25	0.86	1.00										
F6	0.09	0.07	0.05	1.00									
F8	0.12	0.21	0.26	0.40	1.00								
F9	0.25	0.86	1.00	0.05	0.26	1.00							
F10	0.13	0.42	0.43	−0.10	−0.37	0.43	1.00						
F11	0.25	0.87	0.97	0.06	0.26	0.97	0.42	1.00					
F12	0.17	0.21	0.21	0.28	0.28	0.21	0.04	0.22	1.00				
F13	0.01	−0.10	−0.11	0.02	0.08	−0.11	−0.12	−0.10	0.17	1.00			
F14	−0.01	−0.04	−0.06	−0.09	−0.05	−0.06	−0.05	−0.06	−0.22	0.23	1.00		
F15	0.01	−0.05	−0.07	−0.05	0.01	−0.07	−0.07	−0.07	−0.08	0.49	0.74	1.00	
F16	0.07	−0.04	−0.08	0.04	0.04	−0.08	−0.04	−0.08	0.05	0.04	0.02	0.03	1.00

The relationship between the target variable *model_price* and the categorical variables was examined based on the comparison of the target variable average with the values of the categorical features (Figure 3). The mean was chosen as an adequate measure to calculate the average of the model price, due to its robustness against imbalanced data. The results implied that 3D models with vertex colors (*vertex_color*), rigged geometries (*rigged_geometries*), and the option to scale the 3D model (*scale_transformation*) have a higher mean model price than 3D model that do not provide these features. Regarding *uv_layers*, the mean model price seemed to be approximately equal for both values. The variable *pbr_type* had three values. Although the mean model prices of 3D models with the values “None” and “metalness” were comparable, models with the PBR-type specular had a noticeably higher mean value for *model_price*. Lastly, it was apparent that the mean model price varied for the different values of *file_format_score*. While 3D models in up to eight different file formats had a comparable mean model price, models in more than eight file formats had a significantly higher mean model price. The results from the data analysis served as the basis for the selection and engineering of relevant features for the ML models.

5.2.2. Feature Selection

The efficiency and performance of ML models depend on the quality and quantity of the training features [77]. A dataset can include redundant features or features which are irrelevant for the prediction of a response variable and can negatively impact the learning process. Hence, eligible feature must be selected. The three common methods for feature selection are filter, wrapper, and embedded methods [43,45]. Wrapper and embedded methods are intra-learning approaches for feature selection, whereas filter methods can be

used before the training and evaluation of an ML model. The two types of filter methods include univariate and multivariate filter methods, as demonstrated in Section 5.2.1.

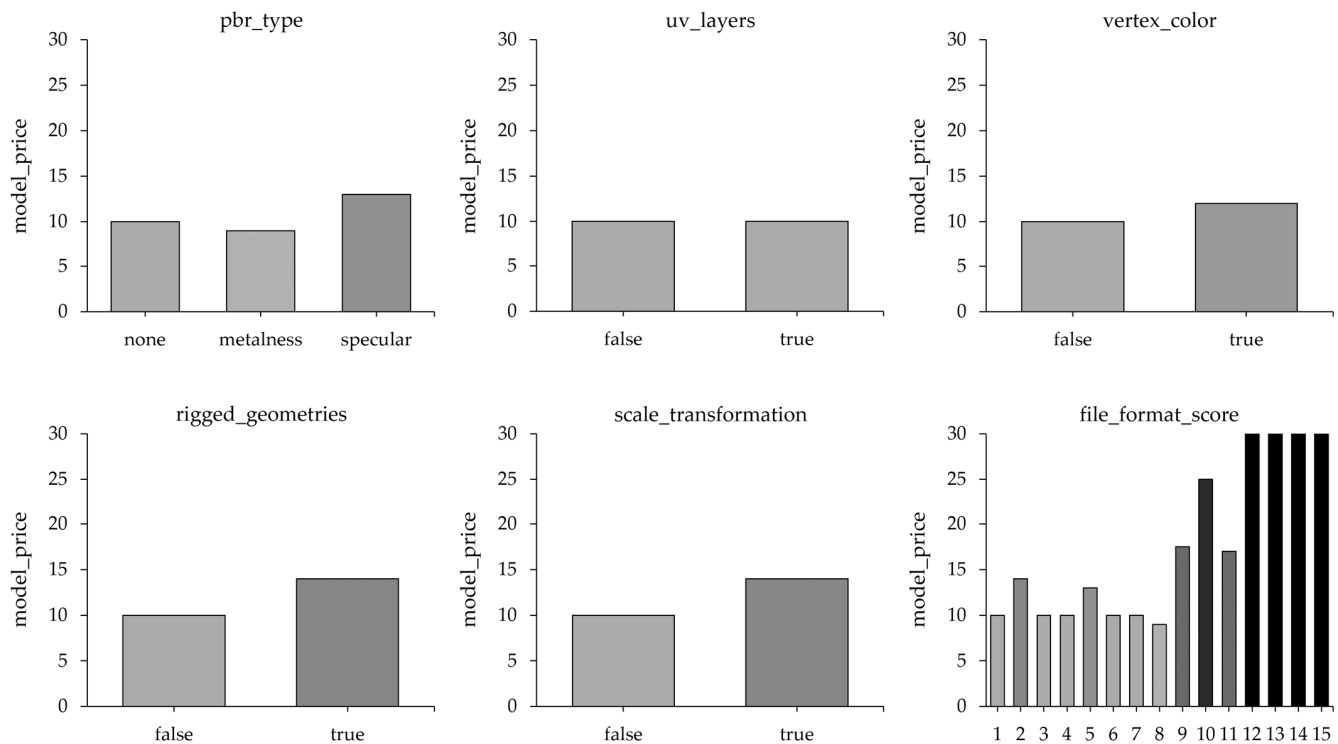


Figure 3. Relationships between the target and categorical variables.

Based on the analysis results, we did not define a correlation threshold for the correlation between the numeric and the target variable *model_price* because most numeric variables were significantly skewed and leptokurtic, as was the target variable. Hence, linear correlations were less conclusive. However, the correlations between the numeric variables allowed the exclusion of features. In terms of the geometric features, the variables *3D_model_size*, *triangles_count*, *face_count*, *polygons_count*, and *quads_count* were redundant because *total_triangles_count* essentially included all information represented in these features and showed a strong correlation with the variables. Therefore, these features were excluded from the feature set. In the appearance-related variables, *textures_count* and *total_texture_size* showed a high correlation. This is not surprising because the aggregated texture file size increased with the number of textures. Hence, *total_texture_size* was excluded as a feature. In the case of categorical features, only the feature *uv_layer* seemed to not influence the price of virtual 3D assets; the average prices of 3D models did not show differences between the inclusion and exclusion of UV layers. Hence, only the *uv_layer* feature was excluded from the categorical features, whereas the others remained in the feature set. The 10 selected features are illustrated in Table 7.

5.2.3. Feature Engineering

The performance of ML models depends on the quality of the input data; therefore, feature engineering is mandatory for accurate results [45]. Feature engineering includes discarding, feature encoding, and feature transformation and scaling methods. To build a robust model and prevent overfitting, features with a high number of low-frequency values must be avoided. This is a common problem with categorical features [45]. In the dataset, none of the features required discarding. However, the categorical features were not numeric. Therefore, it was difficult to compare the variables with numeric features and calculate their relationship with a numeric target variable for regression.

algorithms [45]. Hence, the categorical features were encoded by assigning integer numbers to the categorical values (Table 7).

Table 7. Selected numeric and categorical features for the ML process.

#	Feature	Category	Encoding and Cut-Off Values
F1	model_price	Currency	Cut-off Value: 129
F9	total_triangles_count	Geometry	Cut-off Value: 2.717.542
F12	materials_count	Appearance	Cut-off Value: 46
C1	pbr_type	Appearance	Encoding: none = 0, metalness = 1, specular = 3
F13	textures_count	Appearance	Cut-off Value: 57
F14	textures_mean_sizes	Appearance	Cut-off Value: 36.782,56
C3	vertex_color	Appearance	Encoding: true = 1, false = 0
F15	animations_count	Animation	Cut-off Value: 7
C4	rigged_geometry	Animation	Encoding: true = 1, false = 0
C5	scale_transformation	Configuration	Encoding: true = 1, false = 0
C6	file_format_score	Compatibility	Encoding: 1 format = 1, 2 formats = 2, . . . , 15 formats = 15

Furthermore, because most of the numeric features in the dataset were significantly skewed, a feature transformation was required to ensure normal distribution of the values for the linear regression. To avoid negative values, a logarithmic + 1-transformation was applied to the numeric features. Finally, the value range of the numeric features differed significantly. Hence, normalization of the values was performed to ensure an efficient learning process. Normalization does not affect the distribution of values; therefore, it was applied after the application of the logarithmic transformation [45,74]. Through the normalization process based on the MinMax-scaler [45], the values of all numeric features were scaled in a fixed range of 0 to 1 (for example, *model_price*: Figure 2).

5.3. Model Training and Tuning

For model training, the dataset was split into a test (25% of the entries) and a training dataset (75% of the entries). Models based on multiple linear regression (OLS), ridge regression, and SVR require input data with transformed and scaled values; therefore, two datasets were created, where one was only scaled, and one was logarithmically transformed and scaled. Thus, the comparability of the models could not be ensured regarding all performance metrics, because multiple linear regression, ridge regression, and SVR were applied to a logarithmically transformed set of training data. Therefore, the input values for calculating the metrics were re-transformed after the learning process with an exponential function to ensure comparability of the evaluation results. Before the training of the ML models, a 10-fold cross validation was applied to the training set for validation purpose to facilitate the comparison of the ML model performances. Most of the ML models in this study depended on hyperparameters that influence the quality of the learning process and must be set before the initial learning process [78]. Therefore, a stepwise grid-search approach was conducted to select the hyperparameters, whereby the optimal value for only one hyperparameter at a time was searched. All other parameters were either set to a default or a previously determined optimal value. The models were trained and tested with the different parameters based on cross-validation. Thereby, only those hyperparameter values which exhibited the lowest MAE in the cross-validation process were selected. The results of the hyperparameter search are shown in Figure 4. The evaluation results are illustrated in Table 8.

First, linear and regularized regression models were used to predict prices for virtual 3D assets. The multiple linear regression model was validated using 10-fold cross-validation. Furthermore, the regularized regression models, ridge regression and lasso regression, were trained and evaluated. For the ridge regression, the alpha hyperparameter

was tuned based on a grid-search with 10-fold cross-validation. The alpha parameter was selected from a range of 1 to 200. It was apparent that the MAE gradually increased with an increasing alpha value. Therefore, the lower boundary value of the predefined range for $\alpha = 1$ was selected and the model was evaluated with 10-fold cross-validation. As for the ridge regression model, the alpha hyperparameter for the lasso regression was tuned based on a grid-search with 10-fold cross-validation and the scoring measure MAE. The alpha hyperparameter was selected from a range of 0,1 to 1. The MAE significantly decreased after 0,5, with the lowest value at 1. In turn, $\alpha = 1$ was selected. Subsequently, the model was evaluated with 10-fold cross-validation.

Second, the regression tree models were trained and evaluated. The decision tree was constructed based on its default parameters. Hence, the nodes of the decision tree were expanded until all leaves contain fewer than two samples. The results from the fivefold cross-validation for the decision tree regression are shown in Table 8. After the decision tree regression, ensemble learning models, i.e., random forest regression and extreme gradient boosting trees (XGBoost), were used for the regression task. Random forest regression is based on the decision tree method and combines multiple decision trees via bagging. Therefore, it was vital for the performance of the model to tune the hyperparameter $n_estimators$, which determined the number of trees comprising the forest. For this parameter tuning, grid-search and fivefold cross-validation were deployed, and the $n_estimators$ parameter was selected from a range of 50 to 350. With a growing value of $n_estimators$, the MAE gradually decreased. Although there was a significant improvement between 50 and 150, the improvement flattened beyond this value. However, to ensure robust results, $n_estimators = 350$ was selected. The model was evaluated with fivefold cross-validation. Unlike the random forest approach, the XGBoost method utilizes boosting to create a model based on multiple decision trees and depends on a multitude of hyperparameters, which are essential for the performance of the model. Amongst others, these include the number of decision trees ($n_estimators$), the learning rate ($learning_rate$), the regularization parameter (α), maximum depth of a tree ($maximum_depth$), and the columns sampled by tree ($colsample_bytree$). The lowest MAE values could be identified at $n_estimators = 225$ (range: 25–275), $learning_rate = 0,2$ (range: 0,1–1), $\alpha = 15$ (range: 10–40), and $maximum_depth = 15$ (range: 5–30). In the case of $colsample_bytree$, the MAE significantly decreased until 0.2; then, the improvement flattened beyond these values. Hence, the value was set to $colsample_bytree = 1$ (range: 0,1–1). After the selection of the hyperparameters, the model was trained and evaluated through threefold cross-validation.

Third, the SVR model was trained and evaluated. SVR is a non-linear supervised learning approach for regression analysis that utilizes support vector machines (SVMs) for deriving predictions. SVMs essentially map input data into a high-dimensional feature space; thus, SVR is a computationally expensive approach [79]. Through principal component analysis (PCA), the n -dimensional data were expressed by a smaller number of linear combinations of the features. The aim of PCA is to reduce the dimensionality of the data, while preserving variations. Therefore, PCA was applied to the training set [80]. PCA requires the data to be normalized and standardized [45]; therefore, the logarithmically transformed training set, which had been additionally scaled through MinMax-scaling, was used. SVR could subsequently be applied to the training set with reduced dimensionality. However, before the model could be trained, the hyperparameters needed to be selected. For non-linear tasks, it is advisable to select the radial basis function (RBF) as the kernel function. Furthermore, there were three relevant hyperparameters which optimized the generalization of the model, and thus, the quality of the predictions: C ($=0,7$, range: 0,1–1), γ ($=70$, range: 10–70), and ϵ ($=0,01$, range: 0,01–0,1). These parameters were tuned using grid-search and threefold cross-validation. Subsequently, the SVR model was trained with the selected hyperparameters via threefold cross-validation (Table 8).

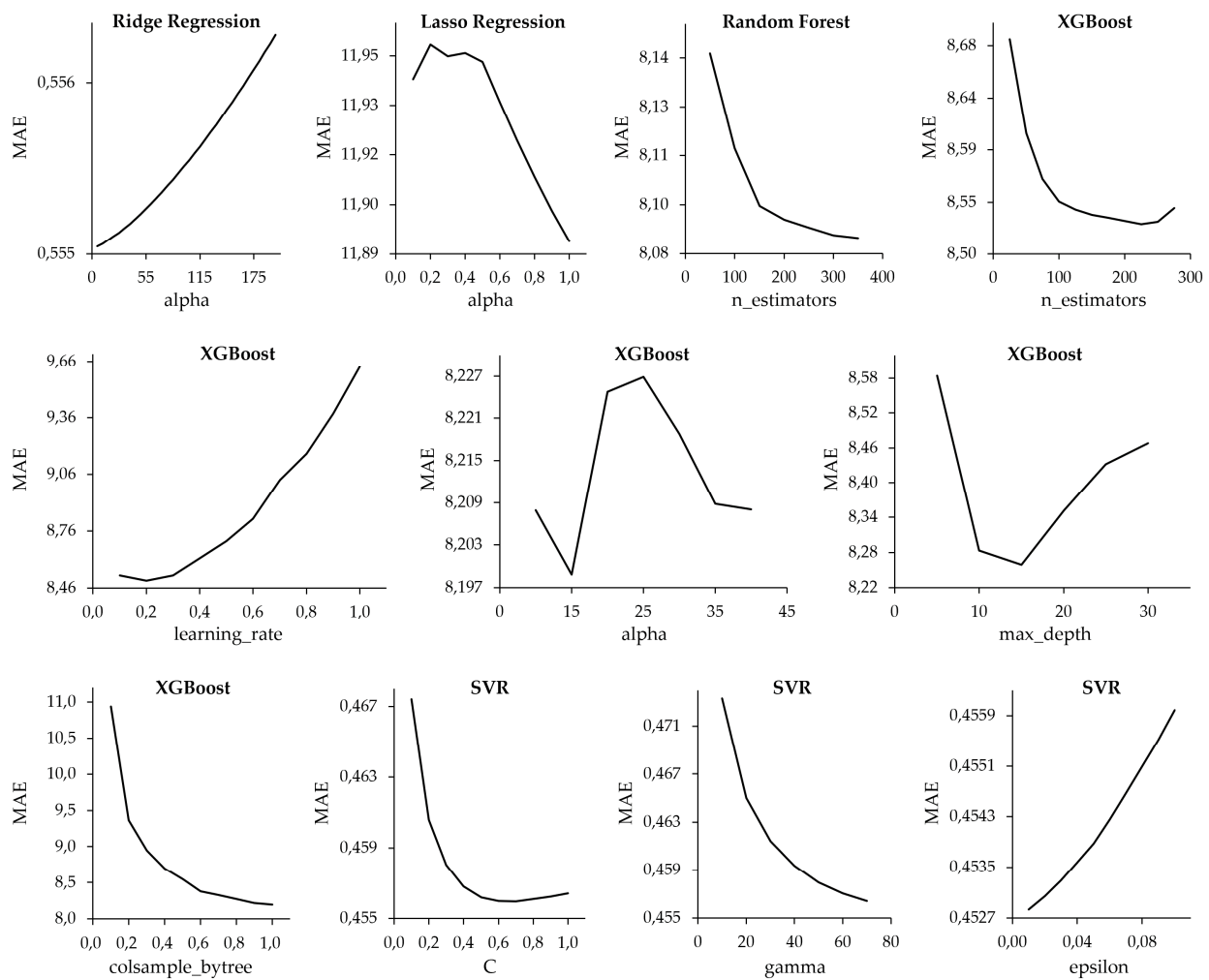


Figure 4. Hyperparameter tuning.

Table 8. Cross-validation results of the ML models.

Model	Measure	MAE	MSE	RMSE	R ²	aR ²
Multiple Linear Regression	Mean	10,640	444,019	21,069	0,146	0,144
	Min	10,361	415,259	20,378	0,136	0,135
	Max	10,876	464,671	21,556	0,152	0,150
Ridge Regression	Mean	10,640	443,999	21,068	0,146	0,144
	Min	10,350	416,388	20,406	0,139	0,137
	Max	10,946	471,101	21,705	0,150	0,148
Lasso Regression	Mean	11,892	375,045	19,363	0,278	0,276
	Min	11,571	351,201	18,740	0,256	0,254
	Max	12,206	398,541	19,963	0,293	0,291
Decision Tree Regression	Mean	10,010	347,412	18,637	0,331	0,331
	Min	9,838	334,153	18,280	0,305	0,304
	Max	10,256	364,986	19,105	0,362	0,362
Random Forest Regression	Mean	8,085	190,706	13,808	0,633	0,633
	Min	8,012	182,984	13,527	0,627	0,627
	Max	8,222	197,737	14,062	0,637	0,637
Extreme Gradient Boosting Trees	Mean	8,150	195,784	13,991	0,623	0,623
	Min	8,128	190,339	13,796	0,615	0,615
	Max	8,215	203,181	14,254	0,630	0,630
Support Vector Regression	Mean	7,467	202,967	14,247	0,610	0,603
	Min	7,454	198,627	14,093	0,598	0,597
	Max	7,489	206,655	14,375	0,621	0,620

6. Evaluation and Demonstration

The results from the training and tuning of the ML models were evaluated in detail to select a final ML model and discuss relevant price determinants. The results were demonstrated and deployed in form of a virtual 3D asset price prediction tool.

6.1. Model Performance

First, the linear regression models were trained and evaluated. The models performed poorly, considering aR^2 scores of 14,4% (multiple linear and ridge regression) and 27,6% (lasso regression). The weak performance in multiple linear regression was expected because the univariate and bivariate data analysis revealed that all numeric features were significantly screwed and none of the features had a significant linear relationship with the target variable *model_price*. Therefore, the multiple linear regression results were considered as a baseline. Ridge regression is especially useful if a dataset has features with undetected outliers, which significantly dilute the performance of the model, whereas lasso can exclude unnecessary features. Therefore, the results of these regression models provided information about the rigor of the outlier handling and feature selection. Both showed weak scores; therefore, the results of the linear regression models are not influenced by features with undetected outliers or irrelevant features. The tree-based models exhibited the best performances. Decision tree regression scored comparably low with an aR^2 of 33,1%, whereas random forest regression showed a significantly higher performance than the single decision tree approach and any of the linear regression models (aR^2 : 63,3%). The XGBoost model provided slightly weaker results than the random forest regression. The mean score during cross-validation for aR^2 was 62.3%, although the scores ranged between 61,5% and 63,0%. Lastly, the SVR showed significantly better results than the linear models with a mean aR^2 score of 60,3%. However, the performance resulted in slightly lower scores compared with the random forest regression and XGBoost models. To conclude, random forest regression model provided the best results for predicting the target variable *model_price*.

6.2. Feature Scoring and Discussion

To gain insights to the relevance of the specific features, and, in turn, the price determinants, the feature importance score (FIS) in the best performing model, random forest regression, was evaluated based on embedded methods [43–45] and the mean decrease impurity (MDI) metric (Figure 5).

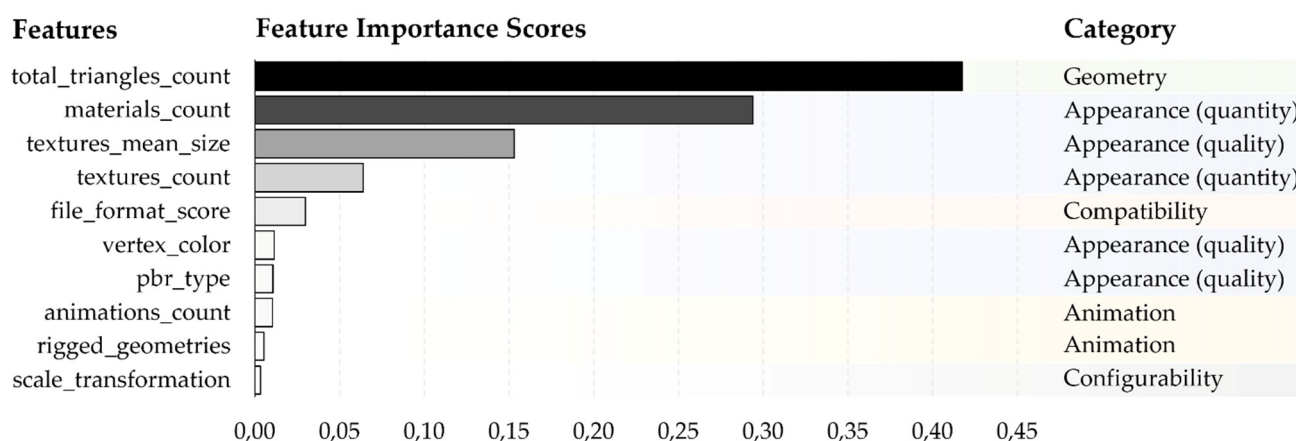


Figure 5. Feature importance scores (FISs) of the random forest regression model.

The scores suggested that the feature *total_triangles_count* had the highest impact on the regression task, with an FIS of 41,76%. In turn, the geometric complexity of a virtual 3D asset was the most important price determinant identified in this study. Hence, virtual 3D asset creators consider the complexity of their geometry as the superior single price

setting criterion. The geometry represented the essence of a 3D model; thus, this result was expected.

However, the importance of the 3D models' appearance was deemed to marginally outreach the geometry-related features considering the feature categories (aggregated FIS "Appearance" = 53,32%). The feature *material_count* (FIS = 29,39%) had the second highest score in the feature importance analysis, followed by the features *textures_mean_size* (FIS = 15,32%) and *textures_count* (FIS = 6,39%). Thus, the number of materials was considered as more important than both the number of textures and their quality combined (FIS = 21,71%). The features *pbr_type* (FIS = 1,08%) and *vertex_colors* (FIS = 1,14%), however, scored comparatively low. Hence, the quantity of the materials, i.e., the number of materials attached to a 3D model, is considered as more important than high-quality materials in the form of PBR materials and vertex colors by the creators. In contrast, the quality of textures in the form of *textures_mean_size* was deemed to be more important to the creators than the quantity of textures (*textures_count*). The results were only partly in accordance with the pricing recommendations of platform providers. The guidelines mention a high quality of textures and the inclusion of PBR materials as important pricing indicators for buyers. Whereas the former was in line with the results from the feature scoring, the latter was not confirmed in the analysis, because creators seemed to only marginally consider PBR materials in their pricing decisions. In addition, creators perceived the number of materials and textures they created as relevant for their pricing decision. Both are not mentioned in the pricing guidelines. However, it must be considered that high numbers of materials and textures have a positive effect on the overall 3D model quality.

Apart from the geometry and appearance of 3D models, the animation-related variables (*animations_count*: FIS = 1,04%, *rigged_geometries*: 0,55%) scored relatively low in the importance ranking (combined FIS "Animation" = 1,59%), although the pricing guidelines suggest that more animations and appropriate rigged models have a positive impact on the purchase decisions. Hence, the creators should reconsider the relevance of animations and rigged geometries in their pricing decisions. The low scoring of animation-related features is surprising because in addition to the recommendations in the pricing guidelines, the rigging and animation of 3D models is a considerably labor-intensive process.

The last two variables concerned the compatibility and configurability of 3D models. The former was represented as the variable *file_format_score* in the dataset, i.e., the number of file formats in which the 3D model was offered. This feature was expected to have a high influence on the pricing decisions because including multiple file formats was described by the platform provider as a vital criterion for purchasers. However, the feature *file_format_score* scores were also comparably low (FIS = 3,00%). Hence, although the number of different file formats was deemed important for the buyers, creators evaluated the relevance of the number of file formats to be significantly lower than the geometry, the appearance, and just slightly higher than including animations and rigged geometries. The reason might be that contemporary computer-aided design (CAD) software allows the export of 3D models in a variety of formats. Hence, the inclusion of multiple file formats is, in most cases, not associated with a high amount of additional work. Although this might explain the low importance score, it is interesting that over 75% of the 3D models only existed in one file format. Thus, although buyers seem to appreciate multiple file formats, and the work intensity to export the model in different formats is comparability low, creators neither consider the number of file formats as vital in their pricing decisions, nor do they include multiple file formats to attract buyers. Finally, the option to configure the 3D model, in this case, to scale the object in form of the variable *scale_transformation*, seems to be irrelevant in the pricing decisions, considering an FIS of 0,33%.

6.3. Price Prediction Tool

Lastly, the results were utilized to implement a tool for virtual 3D asset price predictions. The feature analysis revealed that the feature *scale_transformation* seemed to be irrelevant for the model and training process; therefore, the performance of the random

forest regression model was evaluated again without this feature and compared with the performance of the model containing the full feature set. The mean values of the performance measures from cross-validation for both approaches provide evidence that the full feature set only marginally outperformed the reduced feature set (Table 9).

Table 9. Performance of the random forest regression model with default/reduced feature subset.

Feature Set	MAE	MSE	RMSE	R ²	aR ²
Full Feature Set	8,085	190,706	13,808	0,633	0,633
Reduced Feature Set	8,097	191,305	13,830	0,632	0,632

The dimensionality of the data, and therefore, the complexity of the model, was decreased using the reduced feature set; thus, the reduced feature set was preferred for the training of the final model. Hence, based on the evaluation of the model performance and feature scoring, the random forest regression model with the reduced feature set was chosen for predicting virtual 3D asset prices and implemented in a web-based application (Appendix C, Figure A1). The implementation was based on the web framework Flask [81] and included (1) the selection and deployment of a prediction model previously stored via pickle, (2) the selection of an appropriate HTML template for the user interface (UI), which allows the user to fill in the attributes required for the prediction, and (3) backend implementation which utilizes the provided input data and the ML model to appropriately predict the price of the respective 3D model. When the user has specified and entered all required information about the 3D model, the application predicts its price and states the results to the user as pricing recommendation in the UI.

7. Conclusions and Implications

The objective of this study was to identify relevant price determinants for virtual 3D assets and establish a model that utilizes these determinants for virtual 3D asset price predictions based on a dataset containing the characteristics of 135,384 3D models from the marketplace Sketchfab. To achieve these objectives, data mining and ML-based approaches were deployed, and an exploratory data analysis conducted.

The univariate data analysis provided insights to the values of features and the target variable as well as missing values, data anomalies, and outliers that were excluded. Subsequently, the relationships between the features, as well as those between the features and the target variable, were examined through bivariate data analysis to select appropriate features for the ML process. Thereafter, the features in the selected sub-set were engineered by encoding and transformation to prepare the features for training and tuning of the ML models. Finally, the features were used to train, tune, and evaluate five ML models. For validation purposes, k-fold cross-validation with common error metrics as performance measures was employed. Four out of the five models depended on hyperparameters; therefore, a grid-search was applied to tune the hyperparameters for optimal model performance.

The evaluation revealed that random forest regression is the best performing model for predicting virtual 3D asset prices. To identify the most relevant price determinants in the dataset, feature importance analysis based on the MDI metric was conducted. The feature scoring revealed that the number of triangles, and thus, the geometric complexity of 3D models, is the most important single criterion for model creators (FIS = 41,76%). This result was expected because the geometry is the essence of every 3D model. However, the characteristics regarding the appearance of the 3D model slightly outperformed the geometric-related feature. The number of materials and textures, as well as the texture quality and the inclusion of PBR materials and vertices colors, showed an FIS of 53,32%. Interestingly, the inclusion of high-quality PBR materials seemed to not significantly influence the pricing decision (FIS = 1,08%), although the pricing guidelines of the marketplace providers suggest that PBR materials are highly relevant for the buyers' purchase decisions.

Instead, the creators considered the number of included materials and textures as more important for their pricing decisions (combined FIS = 35,78%). Hence, 3D model creators may reconsider their pricing decisions if they include PBR materials. The same applies for the inclusion of rigged geometries and animations. Although the rigging and animation process is considerably labor-intensive, the creation of rigged geometries and the embedding of multiple animations received little attention in the creators' pricing decision (combined FIS = 1,59%). The buyers appreciate rigged geometries and animations according to the pricing guidelines; thus, creators might reconsider their pricing for rigged and animated 3D models. Lastly, the compatibility of 3D models, represented by the number of file formats in which the 3D model is offered, is less important in the creator pricing decision than expected with an FIS of 3,00%. This finding is remarkable in two aspects. On the one hand, modern CAD software can easily export 3D models in a variety of file formats with little effort. However, more than 75% of the 3D models were only offered in one single file format. On the other hand, the pricing guidelines suggest that compatibility in form of different file formats is of great importance to the buyers. Hence, the creators may consider offering their 3D models in more file formats, and adjust their prices accordingly. Finally, the results from the implementation and evaluation were transferred to a virtual 3D asset price prediction tool based on the random forest regression model which is accessible online (Appendix C, Figure A1).

8. Limitations and Future Research

Although the random forest regression model was tuned to its best possible performance and outperformed all other models, the mean accuracy—represented by the aR^2 score—did not exceed 63%. This result is in line with other studies that have used ML to identify price determinants, and may lie in the fact that the prices for virtual 3D assets are set by the creators who, to some degree, assess their 3D models subjectively. In addition, current pricing guidelines do not provide 3D model creators with information about which 3D model characteristics have the highest impact on the buyers' purchase decisions. Hence, 3D model creators might consider the guidelines but weight the specific characteristics based on their own opinion. Furthermore, this study was limited to the 3D models and characteristics available in the Sketchfab store. For example, Sketchfab do not provide data about the size of material files. Hence, the quality of materials could not be included as a feature. However, the importance of the material quality could be indirectly assessed by including the feature PBR type. In addition, research on virtual 3D asset markets, their pricing, and value is sparse. Hence, the results in this study are based on an explorative approach and the related literature in the domain of ML-based price determinant identification. Lastly, although the results of this study might provide sellers, buyers, and platform providers with insights to the pricing behavior of 3D model creators and the weighted importance of technical 3D model characteristics from the creators' perspective, no information can be derived about the relevant characteristics for buyers, despite the information in the pricing guidelines on the marketplace providers. Hence, future research should investigate the value of 3D models by examining the sales of 3D models regarding their characteristics from the customer perspective. In addition, this study focused on the technical attributes of 3D models. Hence, we know little about other criteria that might influence the pricing and value of virtual 3D assets, such as the actual object represented by the 3D model, its category, or user reviews and ratings. Thus, analyzing the non-technical attributes of 3D models might constitute an interesting future research avenue.

Author Contributions: Conceptualization, J.J.K., U.H.S. and R.Z.; methodology, J.J.K. and U.H.S.; software, J.J.K. and U.H.S.; validation, J.J.K. and U.H.S.; formal analysis, J.J.K. and U.H.S.; investigation, J.J.K. and U.H.S.; resources, J.J.K. and R.Z.; data curation, J.J.K. and U.H.S.; writing—original draft preparation, J.J.K. and U.H.S.; writing—review and editing, J.J.K.; visualization, J.J.K.; supervision, R.Z.; project administration, J.J.K. and R.Z. All authors have read and agreed to the published version of the manuscript.

Funding: We acknowledge support by the German Research Foundation and the Open Access Publication Fund of TU Berlin.

Data Availability Statement: The data presented in this study are openly available in Zenodo at <https://doi.org/10.5281/zenodo.6514727>.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

Table A1. Applied machine learning (ML) models and performance metrics in contemporary publications focusing on ML for price prediction/determinant identification.

Publication	Application	Applied Machine Learning Models	Metrics
[82]	Accommodation	Logistic Regression, Decision Trees/Classification and Regression Tree, K -Nearest Neighbors, Random Forest	AUC-ROC
[49]	Accommodation	Linear Regression, Random Forest, XGBoost	MSE, MAE, R^2
[50]	Accommodation	Linear Regression, Gradient Boosting Machines, Support Vector Machines, Neural Networks, Classification and Regression Trees, Random Forest	MAE, R^2
[51]	Accommodation	Linear Regression, Support Vector Regression, Random Forest	(R)MSE, MAE, (a) R^2
[56]	Cryptocurrency	Logistic Regression, Random Forest, XGBoost, Quadratic Discriminant Analysis, Support Vector Machine, Long Short-Term Memory	Accuracy, Precision, Recall, F1-score
[57]	Cryptocurrency	Linear Regression, Random Forest, Support Vector Machines, Model Assembling	MAE, RMSE, Theil's U^2
[59]	Energy	Gaussian Process Regression, Support Vector Machine, Tree Regression	MAE, RMSE, R^2
[52]	Stock Market	Linear Regression, Elastic Net (Lasso Regression, Ridge Regression), Principal Component Regression, Partial Least Squares, Random Forest, Gradient Boosted Regression Tree, Neural Network, Support Vector Machines	DM test (MSFE), R^2
[53]	Stock Market	Linear Regression, Principal Components Regression, Partial Least Squares, Elastic Net (Lasso Regression, Ridge Regression), Generalize Linear Model, Random Forest, Gradient Boosted Regression Tree, Neural Networks	DM test, R^2
[62]	Warehouse Rental	Linear Regression, Regression Tree, Random Forest, Gradient Boosting Regression Trees	correlation coefficient, RMSE

AUC-ROC: Area under the ROC (receiver operating characteristic) curve; DM test: Diebold and Mariano test; MAE: mean absolute error; (R)MSE: (Root) mean squared error; (a) R^2 : (adjusted) coefficient of determination.

Appendix B

Table A2. Description of regression and machine learning (ML) models applied in this study.

Model	Description
Multiple Linear Regression	Statistical approach for modelling the linear relationship between a dependent variable and one or more independent predictor variables for predicting the former. <i>Ordinary Least Square (OLS):</i> Estimates the parameters or coefficients of the predictor function from the input data, whereas the sign of each coefficient represents the direction of the linear relationship between target and predictor variables. In the case of multiple independent variables: multiple linear regression [83].

Table A2. Cont.

Model	Description
Regularized Linear Regression	Regularization aims to simplify linear regression models through shrinking the coefficient estimates for certain predictor variables by adding a penalty. This adds a bias to the coefficients and results in a lower variance when predicting new data. <i>Ridge Regression</i>: Modified version of the OLS method, aiming to put bias into the estimation of the coefficients to minimize the variance of outcomes. <i>Lasso Regression</i>: Eliminates irrelevant features by imposing a constraint on the model parameters that cause regression coefficients for some variables to shrink towards zero. Given the technical issues regarding concerns on prediction accuracy and interpretation of the OLS, the lasso technique had been proposed to determine a small subset of predictors with the strongest influence from the group of all predictors used [84].
Decision Tree Regression	Predictive ML models for continuous target variables and rule-based techniques that internalize the problem domain including the features of a dataset and their values in a tree-based structure. The features of a dataset are represented as nodes, whereas the observations are modelled as branches of the decision tree. At each node, a rule-based decision on certain feature values is made before the branch is split from the tree, most likely resulting in the best estimate for the target variable. The metric used to find the best partitioning for regression tasks is variance reduction. This decision process is repeated until a leaf node is reached that represents the value of the target variable [83,85]. <i>Decision Tree Regression</i>: Decision trees are easy to implement and—contrary to linear regression—do not pose any special assumption to the dataset; therefore, they do not generalize well, and thus, are prone to overfitting and noise in the dataset.
Random Forest Regression and Extreme Gradient Boosting Trees Regressor	<i>Ensemble Learning</i>: Ensemble learning is utilized to address issues in the decision tree regression. Essentially, multiple decision tree models are combined through bootstrap aggregation or bagging to create one robust ML model. Bagging: Multiple decision trees are trained, each using a random subset of the training dataset. Hence, the target value is not derived from a single decision tree, but rather an average of the predictions from the collection of trees. As a result, it decreases the variance in the overall model, making the ensembled model significantly more robust than the individual models [86]. Additionally, the handling of data becomes easier because the pre-processing of data is less relevant, including the management of missing values [87]. These decision trees are trained on random subsets of data and constitute a forest of decision trees; thus, these models are called random forest [88]. <i>Random Forest Regression</i>: One of the most popular ML models currently used: a supervised ML model which is basically derived from the decision tree model, the random forest model can perform classification and regression tasks [89]. Gradient boosting is a tree- and rule-based, supervised ensemble learning approach. It utilizes gradient descent to minimize a loss function and boosting to enhance the performance of weak instances of the model by retraining them [72,90]. Boosting is an iterative process, which ascribes higher weights to those instances of a model that have exhibited weak predictions, and thus, high error rates. It retrains so-called weak learners sequentially, and thus, learns from previous mistakes, instead of being trained on randomly selected subsets of the data. The error rates of these model instances are used to calculate the gradient, the partial derivatives of the loss function [90]. <i>Extreme Gradient Boosting Trees Regressor (XGBoost)</i>: Scalable algorithm which supports parallel and distributed computing and enhances the performance of the model by identifying more accurate tree models. It computes and utilizes the second-order gradients (or second partial derivatives of the loss function), instead of using the standard loss function as an approximation for minimizing the error of the prediction model. Furthermore, the model applies regularization, which improves its overall generalization, and thus, efficiently prevents overfitting [72]. The chosen learning objective in this study was regression with squared loss.
Support Vector Regression	Kernel-based method based on the work of Cortes and Vapnik [91]: Input data vector is mapped into a high-dimensional feature space. The algorithm learns within this feature space, which is defined by a kernel function. In addition to the standard linear function, there are several kernel functions, such as the radial basis function (RBF) or polynomial function, which can be used for prediction tasks based on non-linear data. Utilizes <i>Support Vector Machines (SVMs)</i> for the regression task, and thus outputs a continuous value. SVMs are extended by the introduction of a tolerance parameter band, which prevents the model from overfitting, and a penalty parameter, which penalizes outliers that are outside of the confidence interval of the kernel function [92].

Table A3. Error and performance metrics for the machine learning model evaluation.

Model	Description
MAE Mean Absolute Error	Average magnitude of variation between the predicted and the observed values. Using only absolute differences in the calculation, residuals with different signs do not cancel each other out. As this is a linear metric, all residuals are weighted equally while calculating the average; hence, the MAE is robust to outliers [93,94].
MSE Mean Squared Error	The MAE calculates the average of the absolute differences between actual and predicted target values, whereas the MSE averages the squared residuals. Despite its popularity, the MSE overestimates the error of a model by squaring the differences between predicted and actually observed values. Therefore, any outliers are penalized significantly [94,95].
RMSE Root-Mean Squared Error	The square-root of the MSE measures the average difference between the predicted and observed values of the target variable. The MSE lacks comparability with the predicted and actual target variable due to it representing the average of the squared residuals. The RMSE mitigates this issue by applying the square-root to the average of the squared residuals. It preserves the units of the target variable; thus, the RMSE allows for improved interpretability. Unlike the MAE and similar to the MSE, the RMSE is sensitive to outliers. Lower RMSE values indicate the better performance of a model [94].
R² Coefficient of Determination	The coefficient of determination is a performance metric for regression models which represent the squared correlation between the predicted and observed target variable. In its essence, it describes the magnitude of variation in the target variable values, which is explained by the predicted values of the latter. In multiple regression models, R ² corresponds to the squared correlation between the observed outcome values and the predicted values of the target variable [96,97]. The R ² is a scale-free metric with values ranging to 1, whereas the performance of the underlying model is positively correlated to the value of R ² . RMSE compares predicted and actual values; however, it does not necessarily provide insights regarding the independent performance of a model. Therefore, it is a more appropriate measure for comparing the errors between two models. Unlike RMSE, the R ² metric can be used to infer the predictive accuracy of a model in percent [98]. However, a weakness of the R ² is that the scores improve with a growing number of predictor variables, thus encouraging overfitting, whereas the model is not improving. A remedy for this issue is in its improvement, the adjusted R ² (aR ²) [99].
aR² Adjusted Coefficient of Determination	The aR ² adjusts the R ² by increasing number of predictors on which the model is trained. Therefore, the score only increases if the additionally added predictor variable is useful. Otherwise, it decreases. The aR ² therefore includes the number of observations (n) and the number of predictor (p) variables used in a model [99].

Appendix C



Virtual 3D Asset Pricing Tool

This pricing tool provides you with a recommendation for your 3D model price based on its characteristics. To allow a price prediction, please fill in the information below:

Geometry	Number of Triangles:		
Texture & Material	Number of Textures:		Does the model include vertex colors: ☐
	Number of Materials:		Does the model include PBR materials: ☐
Animations:	Number of Animations:		Is the 3D model rigged: ☐
File Format:	Number of different 3D file formats:		
Data Size:	Size of all texture files:		Size of all files:

Calculate

Recommended 3D model price:

- \$

Figure A1. Virtual 3D asset price prediction tool.

References

1. Pfouga, A.; Stjepandić, J. Leveraging 3D CAD Data in Product Life Cycle: Exchange—Visualization—Collaboration. In *Transdisciplinary Lifecycle Analysis of Systems Curran*; Wognum, R., Borsato, N., Stjepandić, M., Verhagen, J., Wim, J.C., Eds.; IOS Press BV: Amsterdam, NL, USA, 2015; pp. 575–584.
2. Algharabat, R.; Abdallah Alalwan, A.; Rana, N.P.; Dwivedi, Y.K. Three dimensional product presentation quality antecedents and their consequences for online retailers: The moderating role of virtual product experience. *J. Retail. Consum. Serv.* **2017**, *36*, 203–217. [CrossRef]
3. Hamari, J.; Keronen, L. Why do people buy virtual goods: A meta-analysis. *Comput. Hum. Behav.* **2017**, *71*, 59–69. [CrossRef]
4. Mystakidis, S. Metaverse. *Encyclopedia* **2022**, *2*, 486–497. [CrossRef]
5. Shen, B.; Tan, W.; Guo, J.; Zhao, L.; Qin, P. How to Promote User Purchase in Metaverse? A Systematic Literature Review on Consumer Behavior Research and Virtual Commerce Application Design. *Appl. Sci.* **2021**, *11*, 11087. [CrossRef]
6. Korbel, J.J. Creating the Virtual: The Role of 3D Models in the Product Development Process for Physical and Virtual Consumer Goods. In *Innovation through Information Systems*; Ahlemann, F., Schütte, R., Stieglitz, S., Eds.; Springer International Publishing: Cham, Switzerland, 2021; Volume 46, pp. 492–507. [CrossRef]
7. Unity Asset Store. Available online: <https://assetstore.unity.com/> (accessed on 14 December 2021).
8. MakerBot Thingiverse. Available online: <https://www.thingiverse.com/> (accessed on 14 December 2021).
9. CGTrader Marketplace: The World's Preferred Source for 3D Content. Available online: <https://www.cgtrader.com/> (accessed on 17 December 2021).
10. TurboSquid: 3D Models for Professionals. Available online: <https://www.turbosquid.com/> (accessed on 17 December 2021).
11. About Sketchfab: The Leading Platform for 3D & AR on the Web. Available online: <https://sketchfab.com/about> (accessed on 17 December 2021).
12. Dolonius, D.; Sintorn, E.; Assarsson, U. UV-free Texturing using Sparse Voxel DAGs. *Comput. Graph. Forum* **2020**, *39*, 121–132. [CrossRef]
13. Unity Manual: Materials. Available online: <https://docs.unity3d.com/Manual/Materials.html> (accessed on 17 December 2021).
14. Pai, H.-Y. Texture designs and workflows for physically based rendering using procedural texture generation. In Proceedings of the IEEE Eurasia Conference on IOT, Communication and Engineering (ECICE), Yunlin, Taiwan, 3–6 October 2019.
15. Pan, J.J.; Yang, X.; Xie, X.; Willis, P.; Zhang, J.J. Automatic rigging for animation characters with 3D silhouette. *Comput. Anim. Virt. Worlds* **2009**, *20*, 121–131. [CrossRef]
16. Arshad, M.R.; Yoon, K.H.; Manaf, A.A.A.; Mohamed Ghazali, M.A. Physical Rigging Procedures Based on Character Type and Design in 3D Animation. *Int. J. Recent Technol. Eng.* **2019**, *8*, 4138–4147.
17. 3D Systems: What Is an STL File? Available online: <https://www.3dsystems.com/quickparts/learning-center/what-is-stl-file> (accessed on 10 December 2021).
18. Autodesk: FBX. Available online: <https://www.autodesk.com/products/fbx/overview> (accessed on 10 December 2021).
19. Israel, J.H.; Wiese, E.; Mateescu, M.; Zöllner, C.; Stark, R. Investigating three-dimensional sketching for early conceptual design—Results from expert discussions and user studies. *Comput. Graph.* **2009**, *33*, 462–473. [CrossRef]
20. Park, H.; Moon, H.-C. Design evaluation of information appliances using augmented reality-based tangible interaction. *Comput. Ind.* **2013**, *64*, 854–868. [CrossRef]
21. Riar, M.; Xi, N.; Korbel, J.J.; Zarnekow, R.; Hamari, J. Using augmented reality for shopping: A framework for AR induced consumer behavior, literature review and future agenda. *Internet Res.* **2022**. [CrossRef]
22. Smink, A.R.; Frowijn, S.; van Reijmersdal, E.A.; van Noort, G.; Neijens, P.C. Try online before you buy: How does shopping with augmented reality affect brand responses and personal data disclosure. *Electron. Commer. Res. Appl.* **2019**, *35*, 100854. [CrossRef]
23. Fairfield, J.A.T. Virtual Property. *Boston Univ. Law Rev.* **2005**, *85*, 1047–1102.
24. Lehdonvirta, V.; Wilska, T.-A.; Johnson, M. Virtual Consumerism: Case Habbo Hotel. *Inf. Commun. Soc.* **2009**, *12*, 1059–1079. [CrossRef]
25. Adroit Market Research: Global Virtual Goods Market. Available online: <https://www.adroitmarketresearch.com/industry-reports/virtual-goods-market> (accessed on 11 October 2021).
26. Animesh, A.; Pinsonneault, A.; Yang, S.B.; Oh, W. An Odyssey into Virtual Worlds: Exploring the Impacts of Technological and Spatial Environments on Intention to Purchase Virtual Products. *MIS Q.* **2011**, *35*, 789–810. [CrossRef]
27. Cheung, C.M.K.; Shen, X.-L.; Lee, Z.W.Y.; Chan, T.K.H. Promoting sales of online games through customer engagement. *Electron. Commer. Res. Appl.* **2015**, *14*, 241–250. [CrossRef]
28. Cleghorn, J.; Griffiths, M. Why do gamers buy 'virtual assets'? An insight in to the psychology behind purchase behaviour. *Digit. Educ. Rev.* **2015**, *27*, 91–110.
29. Jiang, Z.; Benbasat, I. Virtual Product Experience: Effects of Visual and Functional Control of Products on Perceived Diagnosticity and Flow in Electronic Shopping. *J. Manag. Inf. Syst.* **2004**, *21*, 111–147. [CrossRef]
30. Ke, D.; Ba, S.; Stallaert, J.; Zhang, Z. An Empirical Analysis of Virtual Goods Permission Rights and Pricing Strategies. *Decis. Sci.* **2012**, *43*, 1039–1061. [CrossRef]
31. Mäntymäki, M.; Salo, J. Why do teens spend real money in virtual worlds? A consumption values and developmental psychology perspective on virtual consumption. *Int. J. Inf. Manag.* **2015**, *35*, 124–134. [CrossRef]
32. Zhu, D.H.; Chang, Y.P. Effects of interactions and product information on initial purchase intention in product placement in social games: The moderating role of product familiarity. *J. Electr. Commer. Res.* **2015**, *16*, 22–33.
33. CGTrader Analytics. Available online: <https://www.cgtrader.com/profile/analytics/about> (accessed on 17 December 2021).

34. CGTrader: What Price Should I Choose for My Models? Available online: <https://help.cgtrader.com/hc/en-us/articles/360015209858-What-price-should-I-choose-for-my-models> (accessed on 6 September 2021).
35. Sketchfab: Seller Guidelines. Available online: <https://help.sketchfab.com/hc/en-us/articles/115004276366-Seller-Guidelines> (accessed on 2 September 2021).
36. TurboSquid: Product Pricing Guidelines. Available online: <https://resources.turbosquid.com/turbosquid/pricing/product-pricing-guidelines/> (accessed on 9 September 2021).
37. Chung, H.M.; Gray, P. Special Section: Data Mining. *J. Manag. Inf. Syst.* **1999**, *16*, 11–16. [\[CrossRef\]](#)
38. Alasadi, S.A.; Bhaya, W.S. Review of Data Preprocessing Techniques in Data Mining. *J. Eng. Appl. Sci.* **2017**, *12*, 4102–4107.
39. Ge, Z.; Song, Z.; Ding, S.X.; Huang, B. Data Mining and Analytics in the Process Industry: The Role of Machine Learning. *IEEE Access* **2017**, *5*, 20590–20616. [\[CrossRef\]](#)
40. Mughal, M.J.H. Data mining: Web data mining techniques, tools and algorithms: An overview. *Int. J. Adv. Comput. Sci. Appl.* **2018**, *9*, 208–215. [\[CrossRef\]](#)
41. Zhu, J.; Ge, Z.; Song, Z.; Gao, F. Review and big data perspectives on robust data mining approaches for industrial process modeling with outliers and missing data. *Annu. Rev. Control* **2018**, *46*, 107–133. [\[CrossRef\]](#)
42. Chu, X.; Ilyas, I.F.; Krishnan, S.; Wang, J. Data cleaning: Overview and emerging challenges. In Proceedings of the 2016 International Conference on Management of data, San Francisco, CA, USA, 26 June 2016.
43. Chandrashekar, G.; Sahin, F. A survey on feature selection methods. *Comput. Electr. Eng.* **2014**, *40*, 16–28. [\[CrossRef\]](#)
44. Jovic, A.; Brkic, K.; Bogunovic, N. A review of feature selection methods with applications. In Proceedings of the 38th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), Opatija, Croatia, 25–29 May 2015.
45. Zheng, A.; Casari, A. *Feature Engineering for Machine Learning: Principles and Techniques for Data Scientists*; O'Reilly Media: Sebastopol, CA, USA, 2018.
46. Nguyen, G.; Dlugolinsky, S.; Bobák, M.; Tran, V.; López García, Á.; Heredia, I.; Malík, P.; Hluchý, L. Machine Learning and Deep Learning frameworks and libraries for large-scale data mining: A survey. *Artif. Intell. Rev.* **2019**, *52*, 77–124. [\[CrossRef\]](#)
47. Schmidhuber, J. Deep learning in neural networks: An overview. *Neural Netw.* **2015**, *61*, 85–117. [\[CrossRef\]](#)
48. Refaeilzadeh, P.; Tang, L.; Liu, H. Cross-Validation. In *Encyclopedia of Database Systems*; Liu, L., Özsu, M.T., Eds.; Springer: New York, NY, USA, 2016; pp. 1–7.
49. Islam, M.D.; Li, B.; Islam, K.S.; Ahasan, R.; Mia, M.R.; Haque, M.E. Airbnb rental price modeling based on Latent Dirichlet Allocation and MESF-XGBoost composite model. *Mach. Learn. Appl.* **2022**, *7*, 100208. [\[CrossRef\]](#)
50. Razavi, R.; Israeli, A.A. Determinants of online hotel room prices: Comparing supply-side and demand-side decisions. *Int. J. Contemp. Hosp. Manag.* **2019**, *31*, 2149–2168. [\[CrossRef\]](#)
51. Chang, C.; Li, S. Study of price determinants of sharing economy-based accommodation services: Evidence from Airbnb.com. *J. Theor. Appl. Electron. Commer. Res.* **2021**, *16*, 584–601. [\[CrossRef\]](#)
52. Drobetz, W.; Otto, T. Empirical asset pricing via machine learning: Evidence from the European stock market. *J. Asset Manag.* **2021**, *22*, 507–538. [\[CrossRef\]](#)
53. Gu, S.; Kelly, B.; Xiu, D. Empirical Asset Pricing via Machine Learning. *Rev. Financ. Stud.* **2020**, *33*, 2223–2273. [\[CrossRef\]](#)
54. Bauer, J.; Jannach, D. Optimal pricing in e-commerce based on sparse and noisy data. *Decis. Support Syst.* **2018**, *106*, 53–63. [\[CrossRef\]](#)
55. Greenstein-Messica, A.; Rokach, L. Machine learning and operation research based method for promotion optimization of products with no price elasticity history. *Electron. Commer. Res. Appl.* **2020**, *40*, 100914. [\[CrossRef\]](#)
56. Chen, Z.; Li, C.; Sun, W. Bitcoin price prediction using machine learning: An approach to sample dimension engineering. *J. Comput. Appl. Math.* **2020**, *365*, 112395. [\[CrossRef\]](#)
57. Sebastião, H.; Godinho, P. Forecasting and trading cryptocurrencies with machine learning under changing market conditions. *Financ. Innov.* **2021**, *7*, 3. [\[CrossRef\]](#)
58. Wang, Q. Cryptocurrencies asset pricing via machine learning. *Int. J. Data Sci. Anal.* **2021**, *12*, 175–183. [\[CrossRef\]](#)
59. Oviedo-Gómez, A.; Londoño-Hernández, S.M.; Manotas-Duque, D.F. Electricity Price Fundamentals in Hydrothermal Power Generation Markets using Machine Learning and Quantile Regression Analysis. *Int. J. Energy Econ. Policy* **2021**, *11*, 66–77. [\[CrossRef\]](#)
60. Roozmand, O.; Nematbakhsh, M.A.; Baraani, A. An electronic marketplace based on reputation and learning. *J. Theor. Appl. Electron. Commer. Res.* **2007**, *2*, 1–17. [\[CrossRef\]](#)
61. Kropp, L.A.; Korbel, J.J.; Theilig, M.-M.; Zarnekow, R. Dynamic Pricing of Product Clusters: A Multi-Agent Reinforcement Learning Approach. In Proceedings of the 27th European Conference on Information Systems (ECIS), Stockholm, Sweden, Uppsala, Sweden, 8–14 June 2019.
62. Ma, Y.; Zhang, Z.; Ihler, A.; Pan, B. Estimating Warehouse Rental Price using Machine Learning Techniques. *Int. J. Comput. Commun. Control* **2018**, *13*, 235–250. [\[CrossRef\]](#)
63. Hevner, A.R.; March, S.T.; Park, J.; Ram, S. Design Science in Information Systems Research. *MIS Q.* **2004**, *28*, 75–105. [\[CrossRef\]](#)
64. Peffers, K.; Tuunanen, T.; Gengler, C.E.; Rossi, M.; Hui, W.; Virtanen, V.; Bragge, J. The design science research process: A model for producing and presenting information systems research. In Proceedings of the First International Conference on Design Science Research in Information Systems and Technology (DESIST 2006), Claremont, CA, USA, 24–25 February 2006.

65. Peffers, K.; Tuunanen, T.; Rothenberger, M.A.; Chatterjee, S. A Design Science Research Methodology for Information Systems Research. *J. Manag. Inf. Syst.* **2007**, *24*, 45–77. [CrossRef]
66. Scrapy: An Open Source and Collaborative Framework for Extracting the Data You Need from Websites. Available online: <https://scrapy.org/> (accessed on 13 June 2021).
67. Pandas. Available online: <https://pandas.pydata.org/> (accessed on 18 June 2021).
68. Harris, C.R.; Millman, K.J.; van der Walt, S.J.; Gommers, R.; Virtanen, P.; Cournapeau, D.; Wieser, E.; Taylor, J.; Berg, S.; Smith, N.J.; et al. Array programming with NumPy. *Nature* **2020**, *585*, 357–362. [CrossRef]
69. Hunter, J.D. Matplotlib: A 2D Graphics Environment. *Comput. Sci. Eng.* **2007**, *9*, 90–95. [CrossRef]
70. Waskom, M. seaborn: Statistical data visualization. *J. Open Sour. Softw.* **2021**, *6*, 3021. [CrossRef]
71. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V. Scikit-learn: Machine learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830.
72. Chen, T.; Guestrin, C. XGBoost. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016.
73. Batista, G.E.A.P.A.; Monard, M.C. An analysis of four missing data treatment methods for supervised learning. *Appl. Artif. Intell.* **2003**, *17*, 519–533. [CrossRef]
74. Pyle, D. *Data Preparation for Data Mining*; Morgan Kaufmann: San Francisco, CA, USA, 1999.
75. Puth, M.-T.; Neuhäuser, M.; Ruxton, G.D. Effective use of Spearman's and Kendall's correlation coefficients for association between two measured traits. *Anim. Behav.* **2015**, *102*, 77–84. [CrossRef]
76. De Winter, J.C.F.; Gosling, S.D.; Potter, J. Comparing the Pearson and Spearman correlation coefficients across distributions and sample sizes: A tutorial using simulations and empirical data. *Psychol. Method.* **2016**, *21*, 273–290. [CrossRef]
77. Pandey, A.C.; Rajpoot, D.S.; Saraswat, M. Feature selection method based on hybrid data transformation and binary binomial cuckoo search. *J. Ambient Intell. Hum. Comput.* **2020**, *11*, 719–738. [CrossRef]
78. Bergstra, J.; Bengio, Y. Random Search for Hyper-Parameter Optimization. *J. Mach. Learn. Res.* **2012**, *13*, 281–305.
79. Thaseen, I.S.; Kumar, C. Intrusion detection model using fusion of PCA and optimized SVM. In Proceedings of the International Conference on Contemporary Computing and Informatics (IC3I), Mysore, India, 27–29 November 2014.
80. Cao, L.J.; Chua, K.S.; Chong, W.K.; Lee, H.P.; Gu, Q.M. A comparison of PCA, KPCA and ICA for dimensionality reduction in support vector machine. *Neurocomputer* **2003**, *55*, 321–336. [CrossRef]
81. Flask: Web Development, One Drop at a Time. Available online: <https://flask.palletsprojects.com> (accessed on 15 July 2021).
82. Afrianto, M.A.; Wasesa, M. Booking Prediction Models for Peer-to-peer Accommodation Listings using Logistics Regression, Decision Tree, K-Nearest Neighbor, and Random Forest Classifiers. *J. Inf. Syst. Eng. Bus. Intell.* **2020**, *6*, 123–132. [CrossRef]
83. Kuhn, M.; Johnson, K. *Feature Engineering and Selection: A Practical Approach for Predictive Models*; CRC Press: Boca Raton, FL, USA, 2019.
84. Tibshirani, R. Regression Shrinkage and Selection Via the Lasso. *J. Royal Stat. Soc. Ser. B* **1996**, *58*, 267–288. [CrossRef]
85. Lewis, R.J. An introduction to classification and regression tree (CART) analysis. In Proceedings of the Annual Meeting of the Society for Academic Emergency Medicine, San Francisco, CA, USA, 22–25 May 2000.
86. Zhang, C.; Ma, Y. *Ensemble Machine Learning: Methods and Applications*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2012.
87. Spedicato, G.A.; Dutang, C.; Petrini, L. Machine learning methods to perform pricing optimization: A comparison with standard GLMs. *Variance* **2018**, *12*, 69–89.
88. Ali, J.; Khan, R.; Ahmad, N.; Maqsood, I. Random forests and decision trees. *Int. J. Comput. Sci. Issues* **2012**, *9*, 272–278.
89. Mohd, T.; Masrom, S.; Johari, N. Machine learning housing price prediction in Petaling Jaya, Selangor, Malaysia. *Int. J. Recent Technol. Eng.* **2019**, *8*, 542–546. [CrossRef]
90. Natekin, A.; Knoll, A. Gradient boosting machines, a tutorial. *Front. Neurobot.* **2013**, *7*, 21. [CrossRef]
91. Cortes, C.; Vapnik, V. Support-vector networks. *Mach. Learn.* **1995**, *20*, 273–297. [CrossRef]
92. Yaohao, P.; Albuquerque, P.H.M. Non-Linear Interactions and Exchange Rate Prediction: Empirical Evidence Using Support Vector Regression. *Appl. Math. Financ.* **2019**, *26*, 69–100. [CrossRef]
93. Brassington, G. Mean absolute error and root mean square error: Which is the better metric for assessing model performance? In Proceedings of the EGU General Assembly Conference, Vienna, Austria, 23–28 April 2017.
94. Chai, T.; Draxler, R.R. Root mean square error (RMSE) or mean absolute error (MAE)?—Arguments against avoiding RMSE in the literature. *Geosci. Model Dev.* **2014**, *7*, 1247–1250. [CrossRef]
95. Bar-Lev, S.K.; Boukai, B.; Enis, P. On the mean squared error, the mean absolute error and the like. *Commun. Stat. Theor. Method.* **1999**, *28*, 1813–1822. [CrossRef]
96. Huang, L.-S.; Chen, J. Analysis of variance, coefficient of determination and F-test for local polynomial regression. *Ann. Stat.* **2008**, *36*, 2085–2109. [CrossRef]
97. Menard, S. Coefficients of Determination for Multiple Logistic Regression Analysis. *Am. Stat.* **2000**, *54*, 17–24.
98. Botchkarev, A. Performance metrics (error measures) in machine learning regression, forecasting and prognostics Properties and typology. *arXiv* **2018**, arXiv:1809.03006. [CrossRef]
99. Srivastava, A.K.; Srivastava, V.K.; Ullah, A. The coefficient of determination and its adjusted version in linear regression models. *Econom. Rev.* **1995**, *14*, 229–240. [CrossRef]