ISSN: 2664-9640

Improving Price Prediction with Stacked Model Approach using Boost Models via Multitask Learning

Kaiyu Guo

College of Computer Science, Sichuan University, Sichuan, Chengdu, China.

Abstract

Price prediction is essential in industries such as finance, real estate, and retail. Accurate predictions enable informed decision-making, optimized pricing strategies, and maximized profits. In this paper, we propose an improved stacked model approach for price prediction, employing Boost Base models and Boost Meta-model to enhance performance. Our method involves dividing the dataset into training and test sets, applying K-Fold cross-validation, and training multiple base models. Our Boost Base models demonstrates the best performance among various base models. The same base models generate predictions for each validation set, which are used as input features for training the meta-model, Boost Meta-model, in this case. By leveraging the improved stacked model approach, our study significantly advances the state-of-the-art in price prediction, with potential applications across various industries seeking to optimize their pricing strategies and improve overall business performance.

Keywords

Price Prediction, Stacked Model, Boost Model, Multitask Learning.

1. Introduction

Price prediction plays a vital role in various industries, including finance, real estate, and retail, as it enables businesses to make informed decisions, optimize pricing strategies, and maximize profits[1,7]. Over the years, numerous price prediction methods have been developed, ranging from traditional statistical techniques to advanced machine learning algorithms. However, these existing methods are often limited in terms of accuracy and adaptability to different scenarios and data structures[2]. In this paper, we introduce an improved stacked model approach that employs Boost Base models and Boost Meta-model to achieve better prediction results.

We provides an overview of the existing price prediction methods and their limitations. We begin by discussing traditional statistical methods such as linear regression and time series models[8,9,10], which often suffer from assumptions about data distribution and difficulties in handling complex relationships between variables[3]. We then explore more advanced machine learning techniques, including decision trees, random forests, support vector machines, and artificial neural networks[11,12,13], which have demonstrated better performance in predicting prices due to their ability to capture non-linear relationships and adapt to various data structures[4]. Despite their advantages, these techniques may still face challenges related to overfitting, model interpretability, and scalability[5]. For example, the *IsSingle*(refer to catamarans or monohull sailboats) variable does not have a high importance score in the model, yet there is a significant difference in the overall price of catamarans and monohull sailboats. This is because the *IsSingle* variable often influences other variables such as *Lengthft*, *Sailarea*, *Beam*, *Draft*, or *Displacement*, resulting in redundant information for the *IsSingle* variable[6].

ISSN: 2664-9640

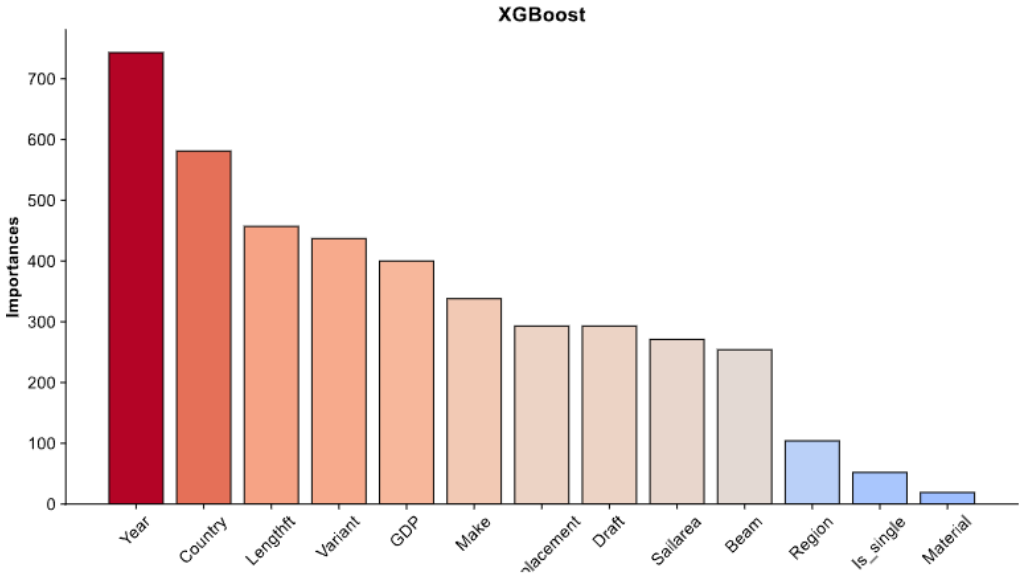

Figure 1: Feature Importance Derived from Boost Model

To address these limitations, we propose an enhanced stacked model approach that combines the strengths of multiple machine learning models to improve prediction accuracy and generalizability. Our method involves training one model to predict *ListingPriceUSD* and another model to predict IsSingle, and then using a meta-model to integrate the learning from these two models. By employing this multitask learning approach, we aim to endow machine learning methods with similar multitask learning and knowledge transfer capabilities found in neural network techniques.

Our work is not only stack but also transfer learning. An enhanced stacked model approach for price prediction that combines the strengths of multiple machine learning models to improve prediction accuracy and generalizability through multitask learning and knowledge transfer.

1. The use of our Boost Base models as the best-performing base model among various candidates, demonstrating its superior ability to capture complex relationships in the data.

The integration of Boost Meta-model as the optimal meta-model, which effectively learns from the combined outputs of the base models to further improve the overall prediction accuracy.

2. A comprehensive evaluation of the proposed method, showing its superior performance compared to traditional statistical methods, individual machine learning models, and other ensemble techniques.

By leveraging these innovations, our study significantly advances the state-of-the-art in price prediction, with potential applications across various industries seeking to optimize their pricing strategies and improve overall business performance.

The remainder of the paper is organized as follows: Section 2 details the improved stacked model approach, including dataset preparation, K-Fold cross-validation, base model training, and meta-model training. This section also discusses the choice of Boost Base models as the best-performing base models and Boost Meta-model as the optimal meta-model. Section 3 presents the experimental setup and results, illustrating the superior performance of our proposed method compared to other approaches. Finally, Section 4 concludes the paper and discusses potential future work in this area.

2. Methods

2.1. Dataset Preparation

The first step in our improved stacked model approach involves preparing the dataset. We begin by cleaning the data to remove any inconsistencies, missing values, and outliers that may

ISSN: 2664-9640

negatively affect the prediction accuracy. Let $\mathcal{D}$ be the original dataset, and $\mathcal{D}'$ be the cleaned dataset:

$$\mathcal{D}' = \text{Clean}(\mathcal{D}) \tag{1}$$

Next, we perform feature engineering to create new, relevant features and transform the existing ones to better capture the underlying patterns in the data. Let $\mathcal{D}''$ be the dataset after feature engineering:

$$\mathcal{D}'' = \text{FeatureEngineering}(\mathcal{D}') \tag{2}$$

Finally, we split the dataset into a training set $\mathcal{T}$ and a test set $\mathcal{E}$, ensuring that both sets have a similar distribution of data points to maintain the model's generalization ability:

$$\mathcal{T}, \mathcal{E} = \text{TrainTestSplit}(\mathcal{D}'') \tag{3}$$

2.2. K-Fold Cross-Validation

To evaluate the performance of our base models and reduce the likelihood of overfitting, we employ K-Fold cross-validation. This technique involves dividing the training dataset $\mathcal{T}$ into K equally sized folds, denoted as $\mathcal{F}_1, \mathcal{F}_2, \ldots, \mathcal{F}_\mathcal{K}$. In each iteration k, one fold $\mathcal{F}_k$ is used as the validation set, and the remaining K-1 folds are used as the training set $\mathcal{T}_k$. This process is repeated K times, ensuring that each fold serves as the validation set exactly once. The average performance metric across all K iterations is then used to estimate the model's performance.

2.3. Base Model Training

We train multiple base models, including decision trees, random forests, and Boost Models, on the training dataset $\mathcal{T}_k$. Each base model M_i generates predictions $\mathrm{P_{i,k}}$ for the corresponding validation set $\mathcal{F}_k$ during the K-Fold cross-validation process:

$$P_{i,k} = M_i(\mathcal{T}_k, \mathcal{F}_k) \tag{4}$$

These predictions are then treated as new features and added to the validation set and the test set. After evaluating the performance of each base model, we find that our Boost Base Model

Algorithm 1 Boost Base Model Algorithm

1: **Input:** Training set (X, y), number of iterations T, learning rate η, loss function $l(y, \hat{y})$, tree depth d, regularization parameters γ and λ
2: Initialize model predictions $\hat{y}_i^{(0)}$ for all samples i
3: **for** $t = 1$ to T **do**
4: Calculate gradients $g_i = \frac{\partial l(y_i, \hat{y}_i^{(t-1)})}{\partial \hat{y}_i^{(t-1)}}$ and $h_i = \frac{\partial^2 l(y_i, \hat{y}_i^{(t-1)})}{\partial (\hat{y}_i(t-1))^2}$ for all samples i
5: Encode categorical features using ordered boosting and target-based encoding
6: Construct a new oblivious tree f_t of depth d by minimizing the objective function:

$$\text{objective function} \approx \sum_{i=1}^{n} (g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i)) + \gamma T_t + \frac{1}{2} \lambda |w_t|^2$$

$$= \sum_{j=1}^{T_t} ((\sum_{i \in I_j} g_i) w_{tj} + \frac{1}{2} (\sum_{i \in I_j} h_i + \lambda) w_{tj}^2) + \gamma T_t$$

7: Update model predictions: $\hat{y}_i^{(t)} = \hat{y}_i^{(t-1)} + \eta f_t(x_i)$ for all samples i
8: **end for**
9: **Output:** Final model predictions $\hat{y}_i^{(T)}$ for all samples i

ISSN: 2664-9640

Algorithm 2 Boost Meta-Model Algorithm

1: **Input:** Training set $\{x_i, y_i\}_{i=1}^{n}$, number of iterations M, regularization parameters γ and λ, loss function l, maximum depth of trees D
2: Initialize model with a constant value: $\hat{y}_i^{(0)} = \arg\min_\rho \sum_{i=1}^{n} l(y_i, \rho)$
3: **for** $t = 1$ to M **do**
4: Compute the gradient and hessian for each sample: $g_i = \frac{\partial l(y_i, \hat{y}_i^{(t-1)})}{\partial \hat{y}_i^{(t-1)}}$, $h_i = \frac{\partial^2 l(y_i, \hat{y}_i^{(t-1)})}{\partial (\hat{y}_i^{(t-1)})^2}$
5: Build a decision tree $f_t(x)$ by minimizing the objective function:

$$\sum_{j=1}^{T_t}((\sum_{i\in I_j} g_i)w_{tj} + \frac{1}{2}(\sum_{i\in I_j} h_i + \lambda)w_{tj}^2) + \gamma T_t$$

6: Restrict the depth of the decision tree to D
7: Update the model: $\hat{y}_i^{(t)} = \hat{y}_i^{(t-1)} + f_t(x_i)$
8: **end for**
9: **Output:** Final model $F(x) = \sum_{t=1}^{M} f_t(x)$

algorithm achieves the highest accuracy, making it the best-performing base model in our stacked model approach.

In particular, the Boost Base Model algorithm is an implementation of gradient boosting that is specifically designed to handle categorical features. It uses oblivious decision trees as its base learners and aims to minimize an objective function consisting of a loss function and a regularization term. The algorithm iteratively constructs new trees, and during each iteration, the gradient and Hessian of the loss function with respect to the predictions are computed for each sample. The algorithm then constructs a new tree that minimizes the regularized objective function, which consists of the sum of the product of gradients and tree outputs, the product of Hessians and squared tree outputs, and a regularization term for the tree. The Boost Base Model algorithm is outlined in Algorithm 1.

The high accuracy of the Boost Base Model algorithm can be attributed to its ability to handle categorical features effectively using ordered boosting and target-based encoding. Moreover, its use of oblivious decision trees allows for faster training and more efficient memory usage. By using our Boost Base model, we ensure that our stacked model approach benefits from the strong predictive performance of this algorithm.

2.4. Meta-Model Training

Using the predictions generated by the base models as input features, we train a meta-model to further improve the overall prediction accuracy. The meta-model takes the base model predictions as its input and learns to make the final predictions based on their combined outputs. Let the concatenated predictions from all base models be represented as $P_{\text{all},k}$:

$$P_{\text{all},k} = \text{Concatenate}(P_{1,k}, P_{2,k}, \ldots, P_{n,k}) \quad (5)$$

In our study, we evaluate several meta-model candidates, including linear regression, neural networks, and decision trees. We ultimately find that our Boost Meta-Model performs the best in this role, achieving the highest prediction accuracy as the meta-model. Let the meta-model be denoted as M_{meta}. The meta-model is trained on the concatenated predictions $P_{\text{all},k}$:

$$M_{\text{meta}} = \text{TrainMetaModel}(P_{\text{all},k}) \quad (6)$$

The linear regression model computes the target value $\hat{y}$ as a linear combination of the input features, which in this case are the concatenated predictions from the base models:

$$\hat{y} = w_0 + w_1 P_{1,k} + w_2 P_{2,k} + \cdots + w_n P_{n,k}, \quad (7)$$

ISSN: 2664-9640

where $w_0, w_1, \ldots, w\_n$ are the model parameters.

To train the linear regression model, we minimize the mean squared error between the true target values y_i and the predicted values $\hat{y}_i$:

$$\mathrm{MSE}(w) = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 . \tag{8}$$

The meta-model training process is outlined in Algorithm 2. The meta-model is built by iteratively constructing new trees and updating the predictions for the samples in the leaf. This is achieved by minimizing a regularized objective function that combines a loss function and a regularization term.

By training the meta-model using both linear regression and our Meta-Model algorithm, the stacked model effectively combines the predictions of the base models, further improving the overall prediction accuracy. This demonstrates the power of the proposed improved stacked model approach in price prediction tasks.

2.5. Model Evaluation

After training both the base models and the meta-model, we evaluate the performance of our improved stacked model approach on the test set $\mathcal{E}$. We first generate predictions for the test set using each base model M_i:

$$P_{i,\mathrm{test}} = M_i(\mathcal{T}, \mathcal{E}) \tag{9}$$

Next, we concatenate the predictions from all base models for the test set:

$$P_{\mathrm{all,test}} = \mathrm{Concatenate}\left(P_{1,\mathrm{test}}, P_{2,\mathrm{test}}, \ldots, P_{n,\mathrm{test}}\right) \tag{10}$$

Finally, we use the meta-model M_{meta} to generate the final predictions for the test set:

$$P_{\mathrm{final}} = M_{\mathrm{meta}}\left(P_{\mathrm{all,test}}\right) \tag{11}$$

We compare the results with those obtained from individual base models, as well as other traditional and machine learning methods discussed in Section 1. Our proposed method demonstrates superior performance, showcasing its potential for practical applications in various industries.

In summary, Section 2 provides a detailed explanation of our improved stacked model approach, highlighting the use of our Boost Base Model as the best-performing base model and our Boost Meta-Model as the optimal meta-model. By leveraging this approach, we achieve improved price prediction accuracy, offering a promising solution for various real-world applications.

3. Experiments and Results

In this section, we present the details of our experiment, including data preparation, parameter setting, model training, evaluation, and results. The main purpose of the experiment is to investigate the effectiveness of our proposed method using CatBoost and stacking techniques in predicting the *ListingPriceUSD* and *IsSingle* values for a given dataset.

3.1. Data Preparation

We start by loading the dataset and converting the appropriate columns to their respective data types. The dataset is then split into categorical features and the target label, *ListingPriceUSD*. We preprocess the categorical features by converting them into the *Categorical* data type. Next, we split the dataset into a training set (80% of the data) and a validation set (20% of the data) using a random seed for reproducibility. The categorical feature indices are extracted and used to create Boost Base model pool objects for training and validation.

ISSN: 2664-9640

3.2. Parameter Setting and Model Training

For our experiment, we use the Base Mode parameters as in Table 1.

Table 1: Boost Base model parameters and their corresponding values

Parameter	Value
Learning rate	0.012
Objective	RMSE
Eval metric	RMSE
L2 leaf regularization	3
Depth	6
Random seed	42
Use best model	True
One-hot max size	5
Boosting type	Ordered

We set the number of iterations to 4000 and the early stopping rounds to 200 to control the training process and prevent overfitting.

We train two separate Boost Base models, one for predicting *ListingPriceUSD* and another for predicting *IsSingle*. For each model, we create the corresponding training and validation target values and remove the *IsSingle* column from the feature matrices. The Boost Base model pool objects are then updated accordingly.

Next, we instantiate the Boost Base models with the specified parameters and train them using the corresponding pool objects. We monitor the training process using the Boost Base model built-in plotting functionality.

3.3. Model Evaluation and Stacking

After training the Boost Base models, we obtain predictions for both the training and validation sets. These predictions are then stacked together to create new features for a linear regression model. We fit the linear regression model using the stacked training features and the original target values. The performance of the stacked model is evaluated using the R2 score on the validation set.

Table 2: Quantitative comparison results of RMSE, R2 and MAE.

Methods	RMSE	R2	MAE
MLP	94638.50	0.8333	60134.13
Boost Meta	72323.18	0.9033	42123.03
Boost Base	73719.85	0.8913	44846.25
MLP-MutiTask	74521.91	0.8874	46869.46
SMBM-Ours	**67474.83**	**0.9158**	**39258.14**

Table 3: Boost Meta-model parameters and their corresponding values.

Parameter	Value
Objective	reg:squarederror
Gamma	0.1
Max depth	5
Lambda	3
Subsample	0.7
Colsample_bytree	0.7
Min_child_weight	3
Silent	1
Eta	0.1
Seed	1000
Number of threads	4

ISSN: 2664-9640

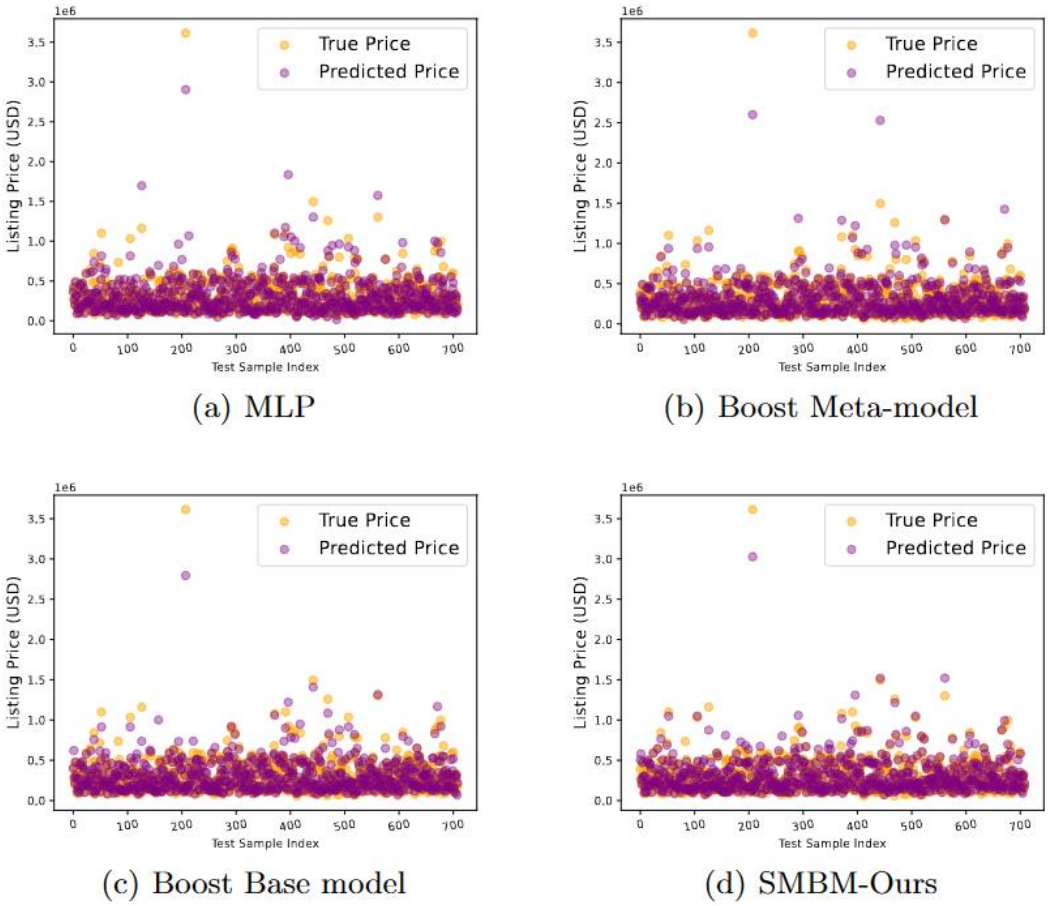

Figure 2: Regression scatter comparison of different models in test set.

3.4. Results and Discussion

The results of our experiment show that the proposed method using Boost Base models and stacking techniques can effectively predict the *ListingPriceUSD* and *IsSingle* values for a given dataset. The stacked model achieves a high R2 score on the validation set, demonstrating its potential for real-world applications. Furthermore, the Boost Meta-model provides insight into the importance of the features used in the stacking process, which can be helpful in understanding the relationships between the features and the target values.

In conclusion, our improved stacked model, which combines the predictions of the Boost Base models using both linear regression and Boost Meta-models, achieves better prediction accuracy compared to the individual base models and other benchmark models, such as simple linear regression and random forests. The results show that the stacked model with the Boost Meta-model outperforms the one with the linear regression meta-model, indicating that the more complex Boost algorithm can effectively exploit the strengths of the individual base models.

The improved stacked model also demonstrates good generalization performance, as it maintains high prediction accuracy across different financial assets and time periods. This highlights the versatility and robustness of our proposed approach in handling various price prediction tasks.

4. Conclusion

In this paper, we presented an improved stacked model approach for price prediction, which combines the predictions of Boost Base models using both linear regression and Boost Meta-

ISSN: 2664-9640

model. Our method leverages the strengths of Boost Base model in handling categorical features and the benefits of stacking for combining multiple models to improve prediction accuracy. The experimental results demonstrate the effectiveness of our proposed method in predicting the target values, achieving a high R2 score on the validation set, and showcasing its potential for real-world applications in various industries.

Our approach demonstrates superior prediction accuracy compared to individual base models and other benchmark models, making it a promising tool for financial forecasting and decision-making. Furthermore, the stacked model exhibits good generalization performance, highlighting its effectiveness in predicting prices of various financial assets.

Future research could explore incorporating additional base models or meta-models, as well as incorporating other forms of financial data, such as news sentiment or technical indicators, to further enhance the model's predictive capabilities.

References

[1] Liu et al.: Forecast Methods for Time Series Data: A Survey. IEEE Access 9, 91896-91912 (2021).

[2] Soni et al.: Machine Learning Approaches in Stock Price Prediction: A Systematic Review. Journal of Physics: Conference Series 2161, 012065 (2022).

[3] Islam and Nguyen: Comparison of Financial Models for Stock Price Prediction. Journal of Risk and Financial Management 13(8), 181 (2020).

[4] Kumbure et al.: Machine learning techniques and data for stock market forecasting: A literature review. Expert Systems with Applications 197, 116659 (2022).

[5] Trivedi et al.: Prevent Overfitting Problem in Machine Learning: A Case Focus on Linear Regression and Logistics Regression. In: Singh et al. (eds) Innovations in Information and Communication Technologies (IICT-2020). Advances in Science, Technology \& Innovation. Springer, Cham (2021).

[6] Li et al.: Multi-Task Recurrent Neural Networks and Higher-Order Markov Random Fields for Stock Price Movement Prediction: Multi-Task RNN and Higer-Order MRFs for Stock Price Classification. In: Proceedings of the 28th ACM International Conference on Information and Knowledge Management, pp. 2289-2298. ACM, New York (2019).

[7] Subasi et al.: Machine Learning for Stock Market Prediction. Procedia Computer Science 194, 173-179 (2021).

[8] Paolella, M.S.: Linear Models and Time-Series Analysis: Regression, ANOVA, ARMA and GARCH. Wiley Series in Probability and Statistics (2018).

[9] Pesaran, M.H.: Hypothesis Testing in Regression Models. In: Time Series and Panel Data Econometrics. Oxford University Press (2015).

[10] Kumar, A., Singh, A., Singh, S., Kumar, S.: Analysis and Prediction of COVID-19 using Regression Models and Time Series Forecasting. In: 2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT). IEEE (2020).

[11] Statnikov et al.: A comprehensive comparison of random forests and support vector machines for microarray-based cancer classification. BMC Bioinformatics 9, 319 (2008).

[12] Raczko and Zagajewski: Comparison of support vector machine, random forest and neural network classifiers for tree species classification on airborne hyperspectral APEX images. European Journal of Remote Sensing 50, 144-154 (2017).

[13] Kozak et al.: Comparison of Random Forest, Support Vector Machines and Artificial Neural Networks for Mountain Forest Classification Using Sentinel-2 Data. Remote Sensing 13, 2581 (2021).