Application of ensemble machine learning methods for modeling the heights of individual forest elements based on inventory data processing

M Vagizov¹*, A Potapov¹, S Navalikhin¹, A Bakeyeva¹ and S Stepanov²

¹Department of information systems and technologies, Institute forest and nature management, St. Petersburg State Forest Technical University, 5 Institutskiy Lane, St. Petersburg 194021, Russian Federation
²Department of applied computer science, Russian State Hydrometeorological University, 79 Voronezhskaya Street, St. Petersburg 192007, Russian Federation

*Corresponding email: bars-tatarin@yandex.ru

Abstract: Machine learning techniques open up new opportunities for research, analysis and can be obtained from forest inventory. Ensemble machine learning methods contain several alternative learning models for predicting characteristics and improving the analysis efficiency of processed data. One of the goals of inventory data analysis is to model heights for individual forest elements, which will allow to build the most accurate models of forest stands growth.

1. Introduction

The class of artificial intelligence methods, machine learning, allows a new approach to processing large amounts of information. A large number of scientific papers are devoted to the topic of data processing using machine learning [1-3]. In Russian forestry one of the cycles of forest inventory works is forest inventory. During forest inventory the forest stand taxation indicators are collected, which, after transferring them to the database, is a very interesting and scientifically large data bank. Different methods of data processing and interpretation, using machine learning allow analyzing the dependencies of the stand. The goals of standard tasks machine learning systems include tasks of typing and clustering of stand features, which allows a deeper understanding of the processes occurring in the forest. Ensemble methods, as one of the main groups of machine learning methods used, allow building more reliable and accurate models based on the processing of large amounts of information. In particular, prepared forest inventory data allow to carry out their processing by different methods and to reveal the most accurate models describing dependencies between taxation indicators.

2. Methods and Materials

Source data structure and import. The test dataset is contained in a txt file. This file represents the results of the Lisinsky forest area inventory conducted in 2005. The information is presented as solid text, there is a formal division into columns, which is broken from time to time: indentation in paragraphs, unusually long data, and in some places the position of elements changes. For this reason preliminary preparation of the data is required before starting the analysis.

The initial dataset is as follows (see figure 1):
Further analysis requires data restructuring and grouping. This task can be divided into three technological stages.

The first stage. Import of unformatted data from the .txt file into a tabular legible form (in this case in .xlsx format).

The second stage. Grouping of processed data so that the characteristics of forest elements are taken into account within the same allotment, while being correlated with other inventory indicators (for correct statistical analysis and machine learning models). For example, it is possible to analyze the indicators of a particular tree species by its age, height, diameter and growth class. For this purpose a multi-index table can be used but in this case duplicating the search results in empty rows would suffice.

The third stage. Excluding some calculated data from the original set in order to optimize subsequent processing: age classes, age groups, food reserves and economic orders.

After completing the higher-level tasks, we get the following data represented in the form of a table 1.

**Table 1.** An example from the dataset prepared for further analysis.

| Numb | Area | N  | Sp  | A  | H  | D  | M  | B  | P  | T  | U  | K  |
|------|------|----|-----|----|----|----|----|----|----|----|----|----|
| 1    | 0.8  | 5  | Spruce | 90 | 27 | 28 | 138| 1  | 0.5 | OX | B2 | 0  |
| 1    | 0.8  | 3  | Aspen  | 100| 32 | 40 | 88 | 1  | 0.5 | OX | B2 | 0  |
| 1    | 0.8  | 2  | Birch  | 100| 27 | 28 | 62 | 1  | 0.5 | OX | B2 | 0  |
| 2    | 3.3  | 4  | Aspen  | 100| 33 | 40 | 130| 1  | 0.6 | OX | C2 | 4  |
| 2    | 3.3  | 2  | Birch  | 100| 28 | 28 | 67 | 1  | 0.6 | OX | C2 | 1  |
| 2    | 3.3  | 2  | Black alder | 100| 28 | 30 | 67 | 1  | 0.6 | OX | C2 | 1  |
| 2    | 3.3  | 2  | Spruce | 100| 27 | 28 | 67 | 1  | 0.6 | OX | C2 | 1  |
| 3    | 2.6  | 7  | Aspen  | 80 | 32 | 36 | 319| 1  | 0.8 | OX | C2 | 4  |
| 3    | 2.6  | 3  | Birch  | 80 | 28 | 26 | 138| 1  | 0.8 | OX | C2 | 1  |

where, the table column names are: Numb – allotment number; Area – the area of the allotment, ha; N - units of forest element; Sp – tree species; A – age of the species, years; H – height, m; D – diameter, cm; M – stock, m³ ha; B – growth class; P – relative completeness; T – type of forest, where OX – oxalis type; U – type of forest growing conditions, K – marketability class.

Category (text) data in the following columns should be replaced with a digital representation: tree species, type of forest and type of forest growing conditions. This is done to optimize work with machine learning models. Substitutions were made for integers: for the column with tree species from 1 to 7, and for the type of forest and forest growing conditions – from 1 to 12, where the most common values were 1, and the rarest were 12.

Prepared and structured data can be imported it into the Jupyter Notebook development environment for further development and configuration of the machine learning model (figure 2).
Figure 2. A complete dataset converted to numeric form in the Jupyter Notebook environment.

The initial dataset includes 3359 allotments that were converted from 206 forestry parcels, taking into account the removed lines using the height filter and data clearing method dropna().

2.1. Machine learning models used for research
The main training models used were ensemble classification and regression methods based on random trees (or Random Forest Classifier (RFC) and Random Forest Regressor (RFR)) from the scikit-learn library. The gradient Boosting method was also tested in two variations: classifier and regressor, but the accuracy of these models was less than that of decision trees. When comparing different methods, the regression methods showed a similar result in terms of accuracy relative to the classifier based on random trees, but the average errors in this case were higher and occurred more frequently in some experiments, the main focus of the analysis was on the RFC model. The KNN method showed low efficiency, less than that of regression methods. For clarity of the study, the model (RFC) will be compared with the model (RFR), which showed less accurate results.

Random Forest Classifier and Random Forest Regressor models were trained on the already prepared dataset from the previous sections of the article. Four main parameters were set up and tested to configure the model: test_size=0.25, n_estimators=100, max_leaf_nodes=500, n_jobs = -1. The first three parameters were determined experimentally, after which the values that show the maximum accuracy with the minimum training time of the model were found.

3. Results and Discussion
Preparation of data. To start working with the data, it is necessary to check it for anomalies and missing values in some cases, so-called noise and outliers.

A very important problem that arises from the method of measuring diameters and heights during stands inventory is the lack of some measurable data for young plantings (sometimes not only for young plantings), which can greatly affect the correctness of the result during data analysis and the implementation of machine learning models.

During the initial visual analysis of the materials it was found that there are outliers that are inherent only where the height of the forest element is less than 5 meters. Also in some cases there are no measurements of diameters or heights.

Therefore, it was decided to study the dataset with the height of the forest element greater than 5 meters. To exclude forest elements where heights, diameters or other important input parameters are missing, the method dropna() is used. It is possible to fill in the missing values using the method fillna, but this method may affect the accuracy of the model's prediction, so it is better to delete rows with missing values.

After visualizing the data of average diameters by height, we get the following (see figure 3):
Figure 3. Graph of the distribution of average diameters by height after excluding outliers and incomplete data.

To verify the data, ensure that there are no anomalies and confirm that the studied stand is a normal forest, the following histograms were created visualizing some grouped data: age, height, diameter and relative completeness (figure 4).

Figure 4. Histograms of the analyzed indicators, where, (a) – age, (b) - height, (c) - trunk diameter, (d) - relative completeness.

It can be seen that there are regularities in the structure of the studied stand: in all cases, the Student's t-distribution is observed, in some cases, the distribution is close to normal. Such data is better suited for analyzing and implementing machine learning models. However, it should be borne in mind that not every stand has a similar structure, for example, young stands or old-growth stands, where there is often a positive or negative asymmetry in the distribution of indicators such as height, diameter, age and other inventory indicators.

Machine learning models test. After validating the dataset, we can start fitting machine learning models. The following parameters were taken as input data (data for training the model): the diameter of the forest element, its age, relative completeness, tree species, type of forest, and type of forest growing conditions (hereinafter referred to as FGC). Forest element is an indicator that characterizes a
clean, single-aged stand consisting of a single tree species. The advantage of the random tree method is that it allows to easily determine the importance degree of the features influence on the predicted indicator (figure 5).

![Feature importance](image)

**Figure 5.** Comparison of indicators by the value of the features influence.

It can be seen that the diameter feature has the highest weight 0.357, followed by age and relative completeness. The remaining features have low weights, but they help to predict heights more accurately. Generally, one input indicator or, for example, two, would suffice, but the indicators that are most important for prediction should be preserved. In this case, if the number of input indicators decreases, the accuracy of the model will deteriorate.

These input features were chosen for training the machine learning model since they can be obtained through visual inventory or from interpreted aerial photographs data.

Estimating the accuracy of machine learning models. After configuring and training the model, it is necessary to evaluate the accuracy of its forecasts. For this purpose the scikit-learn (sklearn) library for Python provides many options for predictive accuracy metrics. It is useful to consider the example of evaluating the quality of a model's forecast using the Random Forest Classifier model. In this case, the prediction accuracy is based on the following functions:

1) accuracy_score is the number of completely correct model predictions divided by the total number of predictions. The higher the percentage, the more accurate the predictive model is. In this case, the accuracy of the model varies from 34 to 37%. This can be interpreted as follows: 34 height values out of 100 predicted are determined without any error.

2) classification_report is a method that shows detailed statistics on errors and predictions for each class of the predicted value. In this case, this is the height value. Next, It is possible to find out in which cases errors occur and how high their values are. The classification_report method will partly help do this (see table 2).

| heights | precision | recall | f1-score | support |
|---------|-----------|--------|----------|---------|
| 5.0     | 0.79      | 0.94   | 0.86     | 16      |
| 6.0     | 0.62      | 0.35   | 0.44     | 23      |
| 7.0     | 0.39      | 0.56   | 0.46     | 16      |
| 32.0    | 0.36      | 0.56   | 0.43     | 9       |
| 33.0    | 0.00      | 0.00   | 0.00     | 2       |
| 34.0    | 0.00      | 0.00   | 0.00     | 1       |
| accuracy|           |        | 0.36     | 2595    |
Precision (accuracy of identification) – the ability of the model to identify only truly positive outcomes from the entire set of positive data labels.

Recall (completeness) – determines the number of truly positive outcomes among all class labels that were defined as "positive".

F1-Score (F1-score) – represents the harmonic average of identification accuracy and completeness, taking into account both indicators.

Accuracy (prediction accuracy) – the ratio of fully accurate predictions to all predicted values. In all cases, the metric value varies from 0 to 1. The higher the value, the better is the quality of the model's prediction.

Analyzing the table and F1-Score indicator (which combines metrics Precision and Recall) It can be seen at which height class errors occurred and how many values belong to each class. No pattern can be traced, but it is possible to determine at what height and its frequency of occurrence errors arise most often.

Cross_val_score is a procedure of cross-section check (cross-validation)(see figure 6).

![Figure 6. Cross-validation procedure.](image)

When using method, the sample is divided into five parts (blocks). First, the model learns from four blocks and evaluates the prediction accuracy on the remaining one block. Then the process is repeated with changing the block on which the model is being tested. This procedure is shown in the table 3.

| Cross-validation result | Split 1 | Split 2 | Split 3 | Split 4 | Split 5 | Mean |
|-------------------------|--------|--------|--------|--------|--------|------|
|                         | 0.30252501 | 0.2858504 | 0.30347785 | 0.30919485 | 0.27965698 | 0.294545 |

Average value of cross-validation accuracy was lower than when using the function accuracy_score, this is due to the fact that each block has a different set of values, which affects the quality of model prediction. The spread of accuracy values is small, which may indicate that the model is not sensitive to the choice of data for training.

In order to have more information about the error value in each predicted case and when using different training models, using a specific code and library function sklearn–predict() a table can be created that shows the error value as a percentage and the number of errors that fall on this value. To do this, the target heights are removed from a test sample and the model is predicted based on the remaining data. After that, the true and predicted values of heights are compared, then the percentage of error is found and distributed according to the specified categories. It is also possible to create cumulative columns with values and their percentages, which allow to evaluate the distribution of errors by their values in more detail.
The sample for checking the error distribution by value was randomly selected using the method `sample()`, the number of predicted values is 1500 (out of 2595). After code execution the resulting dataframe can be represented as the following table 4.

| Errors size | Count | Percentage | Cumulative amount | Cumulative percentage | Count | Percentage | Cumulative amount | Cumulative percentage |
|-------------|-------|------------|-------------------|----------------------|-------|------------|-------------------|----------------------|
| 0%          | 518   | 34.53%     | 518               | 34.53%               | 388   | 25.87%     | 388               | 25.87%               |
| 1-5%        | 488   | 32.53%     | 1006              | 67.07%               | 612   | 40.80%     | 1000              | 66.67%               |
| 6-10%       | 281   | 18.73%     | 1287              | 85.80%               | 338   | 22.53%     | 1338              | 89.20%               |
| 11-15%      | 124   | 8.27%      | 1411              | 94.07%               | 104   | 6.93%      | 1442              | 96.13%               |
| 16-20%      | 56    | 3.73%      | 1467              | 97.80%               | 41    | 2.73%      | 1483              | 98.87%               |
| 21-25%      | 15    | 1.00%      | 1482              | 98.80%               | 11    | 0.73%      | 1494              | 99.60%               |
| 26-30%      | 17    | 1.13%      | 1499              | 99.93%               | 6     | 0.40%      | 1500              | 100.00%              |
| >31%        | 1     | 0.07%      | 1500              | 100.00%              | 0     | 0.00%      | 1500              | 100.00%              |
| \(\sum\)   | 1500  | 100.00%    | 1500              | 100.00%              |

For clarity, data on the percentage of errors and the number of values encountered can be visualized and presented as a histogram (see figure 7).

![Histogram of data analysis results using RFC and RFR ensemble methods.](image)

Analyzing the table, we can conclude that the number of error-free predictions using the method RFC equals to 34.53%. The value for RFR is 25.87%. This means that the regression method predicts unknown heights less frequently and absolutely accurately. Accumulated percentage reveals that 85.5% of the predicted values by the RFC method and 89.2% of heights predicted by RFR method have an error spread from 0% to 10%. For the RFC model 85 out of 100 predicted heights have a margin of error comparable with the data of visual inventory (up to 10%), in 34.53% of cases the values were predicted absolutely accurately.

Despite the large difference in the absolute accuracy of the two models, by the accumulated percentage RFR is relatively more accurate than RFC model starting with an error rate of 6-10% (89.2 and 85.8%, respectively). For large error values RFR turns out to be more accurate than RFC, but in this case absolute accuracy is much more important, because the Random Forest Classifier method is a more accurate solution for predicting heights based on the tested data.
The problem of probabilities and height distribution by diameter. Diameter is an indicator that almost always has a strong correlation with height: in the dataset under study the correlation coefficient is 0.91. Also, the diameter has the highest weight in the importance of features for predicting heights. A sample from the dataset for a specific diameter value, for example, equal to 20 centimeters, visualized as a height distribution histogram, shows that the range of heights encountered is 16 m (see figures 8 and 9).

![Histogram of the height distribution by a diameter of 20 cm.](image)

Figure 8. Histogram of the height distribution by a diameter of 20 cm.

| Height, m | Count |
|----------|-------|
| 12.0     | 0     |
| 14.0     | 1     |
| 16.0     | 2     |
| 18.0     | 5     |
| 20.0     | 3     |
| 22.0     | 7     |
| 24.0     | 1     |
| 26.0     | 2     |
| 28.0     | 1     |
| 30.0     | 0     |

**Figure 9.** Example of a heights table, their number and probability of occurrence with a diameter of 20 cm.

Certainly, the probability of occurrence of some heights is extremely small, but this fact does not allow to accurately predict the height, since with absolutely the same input parameters (diameter, tree species, age, etc.), the height value can vary sometimes to a very large extent (in the example above – 16 meters). This is due to the growth patterns of the stand where under competition or any other external factors trees have different features that differ from the average for the stand in negative or positive directions.

4. **Conclusion**

Despite the problem of the probability distribution of the predicted parameter, the accuracy of the predicted heights remains quite high. If there are other input parameters not used in the study, they can also be used to train the model, thereby increasing the accuracy of the height prediction. During data processing it is possible to predict other unknown parameters apart from height such as classes of marketability. It depends on what input data will be available and what task needs to be solved.

The machine learning models studied in this material can be applied in practice. By combining the methods of ground-based inventory with remote sensing data and using machine learning models it is possible to achieve greater accuracy of data processing results [4-6]. Integrating this results and comparative analysis into inventory process will help to increase its productivity, reduce the cost of manual work and automate some processes if properly harnessed. In particular, the analysis of interpreted features of plantings from aerial photographs and the prediction of the necessary inventory indicators of plantings using machine learning models will allow the formation of databases or inventory descriptions, minimizing interference from the decision-maker.
References

[1] Wang Z et al 2021 Machine learning method for tight-binding Hamiltonian parameterization from ab-initio band structure. npj Computational Materials vol 7(1), 11 doi.org/10.1038/s41524-020-00490-5

[2] Liu Z et al 2018 Application of machine-learning methods in forest ecology: recent progress and future challenges. Environmental Reviews vol 26 (4) pp 339-350 dx.doi.org/10.1139/er-2018-0034

[3] Vagizov M R 2018 Machine learning in the task of forest inventory. Collection of abstracts of the VIII congress of young scientists. Electronic edition. (Saint-Peterburg: ITMO University)

[4] Vagizov M R, Ustyugov V A and Kvochkin D O 2017 Determination of the forest inventory indicators according to the photographs of the unmanned aerial vehicles. Ecology, Environment and Conservation. vol 23 (1) pp 582-586

[5] Istomin E, Petrov Y, Stepanov S, Kolbina O and Sidorenko A 2020 Model of optimum integration of diverse geodata for the benefit of management of forestry. IOP Conference Series: Earth and Environmental Science. 316 012013. DOI: 10.1088/1755-1315/316/1/01201

[6] Tatarnikova T, Stepanov S, Petrov Y, Sidorenko A and Martyn I 2020 A conceptual model for geodata processing for sustainable forest management. IOP Conference Series: Earth and Environmental Science. 507. 012029. DOI: 10.1088/1755-1315/507/1/012029.