Predicting Tree Types in Diverse Ecosystems: A Comparative Analysis of Four Machine Learning Models

Sashwat Rath

University College London

Abstract

In this study, the author develops and compares four distinct classification models to accurately predict tree species in a given area, based on environmental factors such as neighboring tree composition and shadow coverage. Model performance is evaluated using the Macro F1 score as the primary metric. Among the classifiers, K-Nearest Neighbors demonstrates superior performance, while Logistic Regression exhibits the least effective results.

1 Introduction

The Food and Agriculture Organization (FAO) of the United Nations reports a concerning decline in global forest cover, with a reduction of 100 million hectares over the past two decades [1]. Developing models to identify various ground cover types within forests can enable more effective monitoring of these critical ecosystems, facilitating timely interventions to preserve and protect them[2].

This study seeks to develop and assess the performance of four machine learning models - K-Nearest Neighbors, Logistic Regression, Random Forest, and Support Vector Machine - for predicting tree species within diverse ecosystems. Utilizing the Macro F1 score as the primary evaluation metric, we compare the effectiveness of these models in accurately classifying tree types. The results are then analyzed both quantitatively and qualitatively to provide a comprehensive understanding of each model's performance.

2 Background

2.1 Classification Models

2.1.1 Logistic Regression (LR)

LR makes classifications based on the dichotomous result generated from independent variables input in linear regression and the sigmoid function [3, Sec 1]. Elementarily, LR is used for binary classification, while in this research it is applied for multi-classification, which is also known as the Softmax Regression (SR) [**?**]. By applying the Sigmoid function to the basic logic function, LR will have the form as [3, Sec. 2.2]:

$$P(Y) = \frac{1}{1 + e^{-X\beta}} \tag{1}$$

where P(Y) outputs values lying from 0 to 1 [4], $\mathbf{X}$ denotes a vector composed with multiple features of the data and β denotes the parameter vector calculated by the optimization function and will hopefully produce the label to best fit the input. Here, Softmax will be applied to generalize LR to multi-classification problems by computing activation values for each output and normalizing these values to a set of probabilities with the sum of 1, from which label k with the highest probability indicates the predicted label for input x is k [**?**].

2.1.2 Random Forest (RF)

RF is a unique bagging technique that makes use of decision trees. First, the bootstrap approach is used to create m training sets. Each training set is then used to build a decision tree. Instead of choosing the option that maximises the index of all the features when a node discovers a feature that has to be split, it randomly chooses a subset of the features, searches for the best option among those features, and then applies it to the node that needs splitting [5]. The model expression of RF is:

$$\mathrm{y} = \frac{1}{k}\left[\mathrm{t}_1 \cdot \mathrm{prob}(x) + \mathrm{t}_2 \cdot \mathrm{prob}(x) + \ldots + \mathrm{t}_k \cdot \mathrm{prob}(x)\right] \tag{2}$$

where t_i is the i_{th} decision tree, $\mathrm{t}_1 \cdot \mathrm{prob}(x)$ is the prediction of the i_{th} tree for x. The output is the predicted probability of each category (row vector).

$$||\mathbf{x}' - \mathbf{x}_j|| = (\sum_{i=1}^{d} |(x_i)' - (x_i)_j|^p)^{1/p} \tag{3}$$

2.1.3 Support Vector Machines (SVM)

SVM is a supervised learning model that is commonly used for classification and regression problems [6]. The samples can be automatically mapped to an n-dimensional feature space using the Radial Basis Function (RBF) kernel, where n is the number of attributes and each attribute's value is a coordinate value. Following the charting of all the data points, classification is carried out either by a line being drawn or by locating the optimal hyperplane that entirely divides classes [7].The mathematical definition of RBF kernel is as the following [8]:

$$K(x_i, x_j) = exp(-\gamma||x_i, x_j||^2) \quad \gamma > 0 \tag{4}$$

where x_i, x_j are vector points in any fixed dimensional space, and γ is sometimes parameterised using $\gamma = 1/(2\sigma^2)$.

The corresponding minimization problem of SVM is:

$$\min_{\alpha_i} \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\gamma_i\gamma_j\alpha_i\alpha_j exp(-\gamma||x_i - x_j||)^2 - \sum_{i=1}^{n}\alpha_i \tag{5}$$

where $\sum_{i=1}^{n}\gamma_i\alpha_i = 0, 0 \leq \alpha_i \leq C$.

According to the minimization problem of SVM, the minimum value of type depends on the choice of hyperparameters (C, γ).

2.1.4 K-Nearest Neighbours (KNN)

KNN uses the notion of 'distance' to make classifications. Given a data point $\mathbf{x}'$, for which it seeks the classification label, the K 'nearest' $\mathbf{x}$ data points and their known target classes are used to assign a class to $\mathbf{x}'$. The target class is assigned to $\mathbf{x}'$ by computing the class which appears the most amongst the K 'nearest' $\mathbf{x}$ and assigning that particular class to $\mathbf{x}'$. To measure distance in R^d this report uses the $p - norm$ which is the standard Euclidean norm when $p = 2$. [9]

2.2 Performance Metrics

As a commonly used performance metric for dealing with imbalanced datasets, the Macro F1 score was chosen. A nominal categorical variable with k classes ($k \geq 2$) is represented by a $k \times k$ table, which is the confusion matrix. In this table, the columns reflect the actual conditions, while the rows indicate the predicted conditions. The true positive rate (TP_i) is the i-th diagonal element for each class i, followed by FP_i, which is the sum of the i-th row's off-diagonal elements, and FN_i, which is the sum of the i-th column's off-diagonal elements.

The Macro F1 score can be calculated by first finding the precision (P_i) and recall (R_i) as follows [10]:

$$P_i = \frac{TP_i}{(TP_i + FP_i)} = p_{ii}/p_{i.}, \tag{6}$$

$$R_i = \frac{TP_i}{(TP_i + FN_i)} = p_{ii}/p_{.i}. \tag{7}$$

Then the Macro F1 score can be obtained by calculating the harmonic mean of these two [11]:

$$MacroF1 = H(\bar{P}, \bar{R}) = 2\frac{(\frac{1}{n}\sum_x P_x)(\frac{1}{n}\sum_x R_x))}{\frac{1}{n}\sum_x P_x + \frac{1}{n}\sum_x R_x} \tag{8}$$

where H is the harmonic mean.

3 Dataset and Comparison Framework

The dataset contains information about the forest covered in four areas of Colorado. Table 1 shows 12 measures and their descriptions. The first 12 measures are the features for the classifications models and the last measure "Cover Type" is the target classification label.

The last two measures of the features, Wilderness Area and Soil Type, are further expanded into and defined as four and forty binary type variables respectively. In total there are 54 features, with 10 numerical-valued features and 44 binary-valued features and a target class with integer values. Each target class represents a particular type of tree species.

Measurement	Description
Evaluation	Evaluation in meters
Aspect	Aspect in degrees azimuth
Slope	Slope in degrees
Horizontal Distance To Hydrology	Horz Dist to nearest surface water features
Vertical Distance To Hydrology	Vert Dist to nearest surface water features
Horizontal Distance To Roadways	Horz Dist to nearest roadway
Hillshade 9am	Hillshade index at 9am, summer solstice
Hillshade Noon	Hillshade index at noon, summer soltice
Hillshade 3pm	Hillshade index at 3pm, summer solstice
Horizontal Distance To Fire Points	Horz Dist to nearest wildfire ignition points
Wilderness Area(4 binary columns)	Wilderness area designation
Soil Type(40 binary columns)	Soil Type designation
Cover Type(7 types)	Forest Cover Type designation

Table 1: Measurements and Descriptions

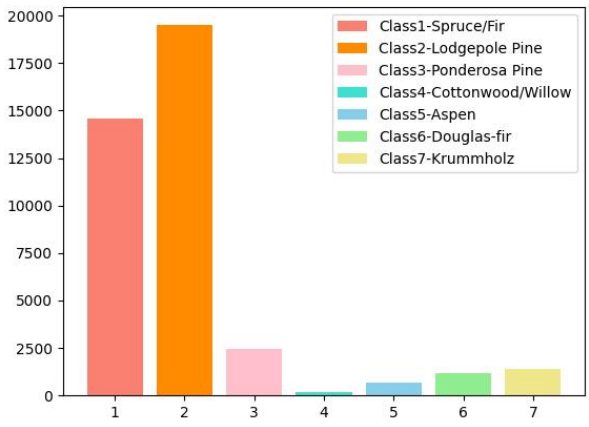

Figure 1: Class Types and The Quantity of Each Class

Figure 1 shows the distribution of data points in terms of the class they belong to. The dataset is heavily imbalanced. Each datapoint $\mathbf{x}$ represents an observation from a 30×30 meter cell [12].

The findings of this paper could help in deciding which areas are most conducive to a particular species' survival, aiming to plant more of the species. Being able to predict the forest cover of an area can allow making inferences about the area.

Over a period of time, such information could be useful as well. Observations and predictions starting to drastically differ could indicate a qualitative change in the area.

Larger inferences can also be made from this classification task. Extending the test sample to other areas of the world with similar features as described by the dataset, one could predict what type of flora would grow in other similar areas of the world.

4 Method

This report aims to apply LR, RF, SVM and KNN to a sample of the dataset and analyse their performance according to the Macro F1 metric. A comparison is then done to choose the best-performing model. This raises interesting questions as to which model performs the best on this dataset to be able to be used for the aforementioned reasons in section 3. The best-performing model also gives insights into the model and dataset which will be discussed qualitatively later in the paper.

This section demonstrates the method of the whole procedure. The sampling method and data representation are described, after which model selection is discussed. Based on this the hyperparameters of the four models used in this research are described.

4.1 Data Sampling

40,000 data points were first sampled from the original 581,012 data points. As seen in section 3, the dataset is heavily imbalanced. To preserve the structure of the dataset, stratified sampling was carried out. For stratified sampling the dataset is divided into groups (strata). Then, from each strata simple random Tsampling occurs according to the proportion in which the strata exists in the dataset. The sample is then produced. This method is known as stratified sampling [13]. The instances were split into stratas corresponding to their class labels and datapoints were randomly sampled from each strata.

After sampling, the feature 'Soil_Type15' is noticed to contain all zeros and so is removed as it would have no impact on the models. This created a sample with 53 features. Then, column-wise normalisation was carried out to scale data, creating the final sample dataset.

4.2 Data Representation

The original dataset is represented by $\mathbf{X}_0$. The final sample dataset is represented as a design matrix and a target matrix. The design matrix $\mathbf{X}$ contains 17,500 data points and 53 features and has shape (N, D) where $N = 17,500$ and $D = 53$. Each row in the design matrix represents one datapoint $\mathbf{x}_n$ where:

$$\mathbf{x}_n = (x_n^1, x_n^2, x_n^3, \cdots, x_n^D) \tag{9}$$

and x_n^i is the ith feature of the nth datapoint, $i \in Z$ and $n \in R$. The ith feature of a datapoint is extracted from the ith column of the datapoint from the sample. The design matrix X is then defined as:

$$\mathbf{\Phi} = (\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \cdots, \mathbf{x}_N)^T \tag{10}$$

Then each column of the design matrix can be represented by:

$$\boldsymbol{\phi}^i = (x_1^i, x_2^i, x_3^i, \cdots, x_N^i)^T \tag{11}$$

for $i \in [1, D]$, where $\boldsymbol{\phi}$ has components which are numerical real valued for $i \in [1, 10]$ and $\boldsymbol{\phi}$ has components which are binary valued for $i \in [11, D]$. These $\boldsymbol{\phi}^i$ are the feature vectors.

The target matrix is defined as follows:

$$\mathbf{T} = (t_1, t_2, t_3, \cdots, t_N)^T \tag{12}$$

where t_j is the target label for $\mathbf{x}_n$, when $j = n$ for $j, n \in [1, N]$

4.3 Model Selection

Model selection is an important step as the performance of a model can be greatly affected by the choice of hyperparameters. In this report, grid search and cross validation are used for selecting models.

4.3.1 Grid Search

Grid search is an effective method of selecting the model with the best performance. It helps models enumerate every possible hyperparameter combination to find the best one. The dimension of the grid is the number of hyperparameters. If there are k hyperparameters and each one has m candidates, then we have to traverse k^m combinations.

Additionally, grid search algorithms normally should be judged by performance criteria such as cross-validation [14].

4.3.2 Cross Validation

Cross validation is a useful method for evaluating the performance of machine learning models [15]. It is often used in situations where the amount of data is limited. This allows it to train and evaluate the model using a variety of subsets of the data, which can provide a more accurate assessment of the model's performance. If the data is divided into 10 folds, for instance, the model will be trained on 9 folds and assessed on the 10th fold. Up until all 10 folds have been used for both training and evaluation, the aforementioned procedure will be repeated [16]. Therefore, it helps to ensure that the model is not overfitting to the training data, and that it is able to generalize well to unseen data [17].

4.3.3 Hyperparameters

The following describes the hyperparameters for each model.

Logistic Regression: This research has applied the built-in functions to fit and optimize the LR model by regularization. The parameters include *penalty*, *solver* and *ll_ratio*. They respectively represent the

regularization function applied, optimization algorithm used and combination ratio of the penalty.

Random Forest: This research chooses *n-estimators* from framework hyperparameters and *max-leaf-nodes* and *max-depth* from decision tree hyperparameters. *n-estimators* specifies the number of classifiers in the random forest. *max-leaf-nodes* are the lowest level nodes in the decision tree. *max-depth* of a decision tree is the maximum length from the root nodes to the leaf nodes.

K-Nearest Neighbours: The K-Nearest Neighbours model has only one parameter, K. The parameter K controls the the number of nearest $\mathbf{x}_j$ to choose and hence controls the degree of smoothing. Since KNN has no 'training phase' average Macro F1 were calculated for each K over different folds after which gridsearch was carried out.

Support Vector Machines: There are two important hyperparameters in the SVM, C and γ. C is the penalty parameter, which trades off the complexity of the support vector against the misclassification rate in the optimization function and can be understood as a regularization factor. The RBF kernel function's coefficient γ expresses how one sample affects the categorization hyperplane as a whole.

4.4 Performance Criterion

Macro F1 score is used as the performance criterion and the four classification models are compared based on it.

This metric was selected because it treats all classes equally and produces superior results when dealing with imbalanced datasets. The Macro F1 score takes into account the model's performance on all classes, not only the classes that performed well in the dataset, by computing F1 values for each class and then average them.

Additionally, compared to other metrics like accuracy, it is less sensitive to the overall distribution of classes in the dataset, which is another factor in its selection. This may be particularly applicable when the class distribution is imbalanced and not appropriate to use accuracy as a performance metric.

5 Evaluation

5.1 Experimental Settings

To get better results, pre-processing was performed before the experiment. The data was sampled and processed as described in Section 4.1. The next step focuses on model selection using grid search and cross validation. As shown in figure 2. For LR, RF and SVM, at first, the $\mathbf{X}$ is split into a train dataset $\mathbf{X}_{TR}$ and a test dataset $\mathbf{X}_{TE}$ to the ratio of 7:3. $\mathbf{X}_{TR}$ is applied to the cross validation with each possible hyperparameters' combination selected in the grid search. Thirdly, the model with best hyperhyperparameters is selected and the $\mathbf{X}_{TR}$ is applied to train this model. At last, $\mathbf{X}_{TE}$ is used to get the final score as the input of the trained model. For KNN the procedure described in section 4.3.3 is used.

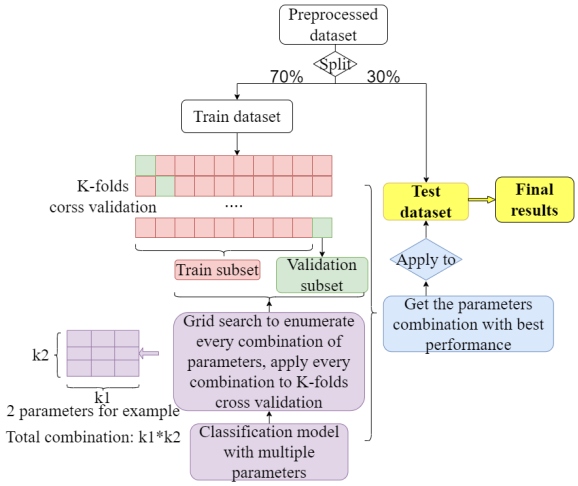

Figure 2: Process of grid seach+CV

Through gridsearch and cross-validation the final optimal hyperparameters for each model were determined as follows:

Logistic Regression: The best value for the hyperparameter *ratio* is 1.0, which will produce the highest Macro F1 score of 0.48.

Random Forest: In the Random Forest model, the hyperparameter combination is: $max\_leaf\_nodes = 2000$, $n\_estimator = 100$, $max\_depth = 40$.

When $max\_depth \in [0, 40]$, $max\_depth \propto$ Macro F1 score. Otherwise, $1/max\_depth \propto$ Macro F1 score. This means that if max_depth is too large, the model will overfit. When we do not limit the maximum depth it has infinite flexibility and can keep growing until it generates a leaf node for each individual observation to achieve perfect classification. At this point, while the model is error-free on the training data, it may show large errors for the test set.

K-Nearest Neighbours: In this experiment Macro F1 scores were averaged out over 10 folds to provide an average Macro F1 score to evaluate the model at a particular K value. A gridsearch was then carried out to find the optimal K value for *KNN*. $K = 1$, the nearest-neighbour rule, was found to be the value of K which provided the model with the highest Macro F1 score of 0.73.

Support Vector Machines: In the experiments with the SVM model, the RBF kernel was first selected based on the features of the data and the aim of the experiment. Then the best-performing model

was found by grid search and cross validation when C = 10 and gamma = 1.

5.2 Results

Figure 3 shows the Macro F1 score of evaluated models (x-axis). As shown in the picture, KNN achieves the highest F1 score with 0.73 while LR gains the lowest score with 0.48.

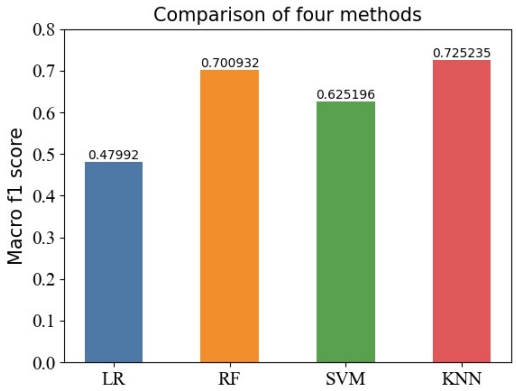

Figure 3: Comparison of four methods' Macro F1 score

Figure 4 shows that the prediction results of the data are approximately the same for different models revealing that the prediction error is mainly due to the imbalanced distribution of the data other than models. Also it can be found in the first and second categories which are lighter in colour in the heatmap that the data sample has a higher amount of data.

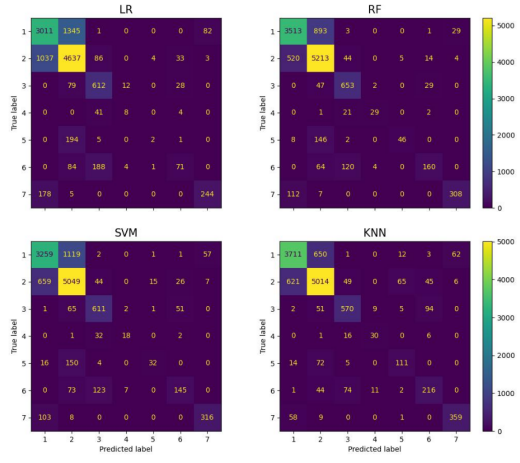

Figure 4: Confusion Matrix Heatmaps

6 Discussion and Conclusion

6.1 Logistic Regression

LR optimizes the model parameter β by iteratively minimizing the cross-entropy loss function $J(\beta)$, meaning to partially derivate the function with respect to β and update β with $\beta_1 = \beta_0 - \alpha\frac{\partial J(\beta)}{\partial \beta}$ until it reaches the best β. In this optimization process, categories with more samples tend to influence β more. In this research, unevenly distributed data is dealt with, thus leading to poor accuracy due to that some smaller number of classifications contribute less to the optimization.

6.2 Random Forest

Figure 3 shows that RF gains the second highest Macro F1 score. RF uses an integrated algorithm which is inherently more accurate than most individual algorithms. Moreover, it increases the random noise in the sample space, parameter space and model space, which reduces the influence of 'individual cases' and further improves the generalization ability.

6.3 Support Vector Machines (SVM)

By first correcting the empirical risk and then minimising the confidence risk, SVM can perform more effectively. SVM's RBF kernel function, however, struggles to handle data sets with biased classes. Data standardisation addresses this limitation. Additionally, it has been demonstrated that SVM is ineffective at classifying imbalanced data. This can be enhanced by undersampling, which includes adding new samples by randomly choosing from a smaller number of the majority class samples, and oversampling, which entails adding samples by randomly choosing from a larger number of the minority class samples. However, under-sampling data under-utilizes and fails to learn the properties of the entire model, while oversampling data is likely to result in over-fitting.

6.4 K-Nearest Neighbour (KNN)

The nearest-neighbour rule was found to be the optimal parameter for KNN implying that t_j is very sensitive to a change in $\mathbf{x}_j$. It was also found that $1/K \propto$ Macro F1 score holds which also illustrates the argument mentioned above. KNN is a simple model has no 'training phase' allowing one to add new data to the model without having to retrain it as the new data wont affect the model. Since the data has 53 dimensions, the calculation of the metric is complex, which also may have resulted in reduced performance. Considering the data's sensitivity this model's simplistic approach gives it the highest score.

6.5 Conclusion and Future work

This study has thoroughly evaluated the performance of four distinct classification models on a selected dataset, employing both qualitative and quantitative analyses. Notably, the simplest model, K-Nearest Neighbors, outperforms the others with a Macro F1 score of 0.73, while Logistic Regression yields the lowest score at 0.48. Random Forest and Support Vector Machine also demonstrate relatively strong performance, with Macro F1 scores of 0.70 and 0.63, respectively.

Future research could explore the impact of employing equal sample sizes from each stratum. Although this approach may introduce bias due to undersampling certain strata, the potential trade-off for enhanced model performance merits investigation. Additionally, incorporating neural networks to introduce more complex learning layers may further improve model accuracy and overall performance.

Declaration

This represents the group report submission for group 9 in fulfillment of the project assignment on the module Foundations of Machine Learning (INST0060). In submitting this document, the authors certify that all submissions for this project are a fair representation of their own work and satisfy all UCL regulations.

References

[1] T. Payn, J.-M. Carnus, P. Freer-Smith, M. Kimberley, W. Kollert, S. Liu, C. Orazio, L. Rodriguez, L. N. Silva, and M. J. Wingfield, "Changes in planted forests and future global implications," *FOREST ECOLOGY AND MANAGEMENT*, vol. 352, pp. 57–67, SEP 7 2015.

[2] V. Kiyko, V. Lytvyn, L. Chyrun, S. Vyshemyrska, I. Lurie, and M. Hrubel, "Forest cover type classification based on environment characteristics and machine learning technology," in *International Conference on Data Stream Mining and Processing*. Springer, 2020, pp. 501–524.

[3] D. W. Hosmer Jr, S. Lemeshow, and R. X. Sturdivant, *Applied logistic regression*. John Wiley & Sons, 2013, vol. 398.

[4] C. A. U. Hassan, M. S. Khan, and M. A. Shah, "Comparison of machine learning algorithms in data classification," in *2018 24th International Conference on Automation and Computing (ICAC)*. IEEE, 2018, pp. 1–6.

[5] L. Breiman, "Random forests," *MACHINE LEARNING*, vol. 45, no. 1, pp. 5–32, OCT 2001.

[6] V. Vapnik, *The nature of statistical learning theory*. Springer science & business media, 1999.

[7] M. A. Aizerman, "Theoretical foundations of the potential function method in pattern recognition learning," *Automation and remote control*, vol. 25, pp. 821–837, 1964.

[8] J. Vert, "A primer on kernel methods. kernel methods in computational biology, b. schölkopf, k. tsuda and j,-p vert," 2004.

[9] O. Kramer, *K-Nearest Neighbors*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 13–23. [Online]. Available: https://doi.org/10.1007/978-3-642-38652-7_2

[10] K. Takahashi, K. Yamamoto, A. Kuchiba, and T. Koyama, "Confidence interval for micro-averaged f1 and macro-averaged f1 scores," *Applied Intelligence*, vol. 52, no. 5, pp. 4961–4972, 2022.

[11] J. Opitz and S. Burst, "Macro f1 and macro f1," *arXiv preprint arXiv:1911.03347*, 2019.

[12] D. A. Jock A. Blackard, Dr. Dean. (1998) old_covtype.info. Colorado State University. [Online]. Available: https://archive.ics.uci.edu/ml/machine-learning-databases/covtype/

[13] M. P. Cohen, *Stratified Sampling*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 1547–1550. [Online]. Available: https://doi.org/10.1007/978-3-642-04898-2_574

[14] C.-W. Hsu, Chang *et al.*, "A practical guide to support vector classification," 2003.

[15] G. Seni and J. F. Elder, "Ensemble methods in data mining: improving accuracy through combining predictions," *Synthesis lectures on data mining and knowledge discovery*, vol. 2, no. 1, pp. 1–126, 2010.

[16] G. J. McLachlan, K.-A. Do, and C. Ambroise, "Analyzing microarray gene expression data," 2005.

[17] G. C. Cawley and N. L. Talbot, "On over-fitting in model selection and subsequent selection bias in performance evaluation," *The Journal of Machine Learning Research*, vol. 11, pp. 2079–2107, 2010.