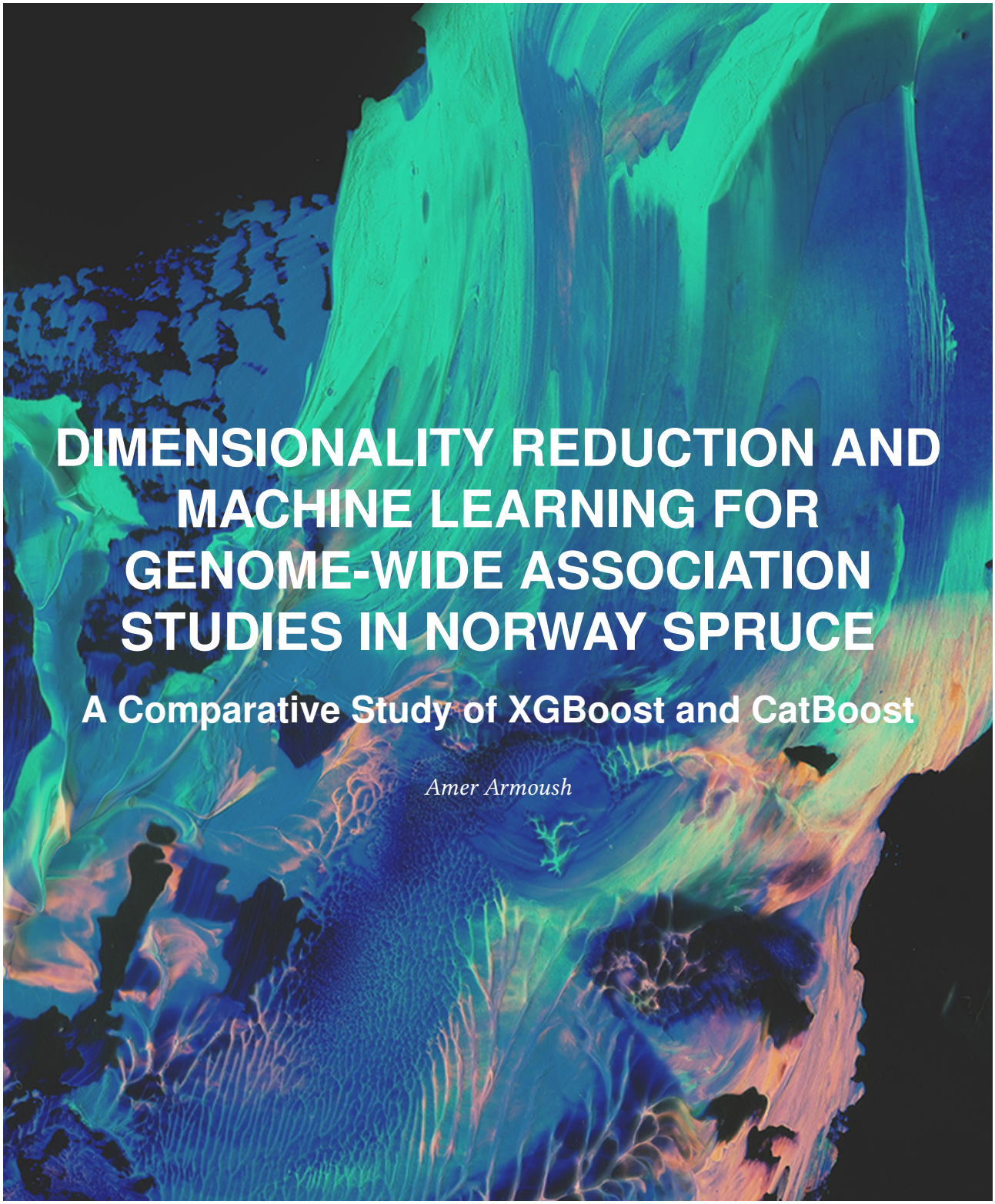
UMEÅ UNIVERSITY



DIMENSIONALITY REDUCTION AND MACHINE LEARNING FOR GENOME-WIDE ASSOCIATION STUDIES IN NORWAY SPRUCE

A Comparative Study of XGBoost and CatBoost

Amer Armoush

Bachelor Thesis, 15 hp/credits
Kandidatprogrammet i Datavetenskap

2025

External Supervisor
María Rosario García Gil

Internal Supervisor
Gustav Grund Pihlgren

Examiner
Suna Bensch

Abstract

Genome-wide association studies (GWAS) are studies in the field of genetics which are continuously conducted on genetic data to find correlations between particular gene markers and diseases or genetic variations. The specific genetic variation that are commonly examined are single nucleotide polymorphisms (SNPs). Machine learning models, which gained a lot of popularity over the last decade in the genetics field have shown a potential in identifying and understanding these genomic correlations. However, the accurate prediction of machine learning models when trained on high-dimensional genomic data presents a significant challenge in the field of supervised machine learning, necessitating the evaluation of various regression models and approaches to identify the most effective approach. This thesis utilizes two prominent models: Extreme Gradient Boosting (XGBoost) regression Catagorical Boost (CatBoost) regression, aiming to compare and analyze the results to determine how dimensionality reduction via Principal Component Analysis (PCA) influences their performance and interpretability. The research focuses on Norway Spruce genomic data, employing cross-validation and hyperparameter tuning to assess prediction accuracy and computational efficiency, and uses the Shapley Additive Explanations (SHAP) package to explain the output of the models.

The findings reveal a critical trade-off between computational efficiency, predictive accuracy, and model interpretability. PCA drastically reduced training times by over 98% for both models. XGBoost proved highly robust to dimensionality reduction, maintaining stable predictive accuracy (RMSE) across all dataset configurations. In contrast, CatBoost's accuracy declined more noticeably on PCA-reduced datasets but achieved the best absolute performance on the original data. A key difference was found in model interpretability: SHAP-based feature elimination showed that CatBoost attributed predictive power to a minimal set of SNPs (as few as 4-5 for some traits), suggesting a simple genetic architecture. On the other hand, XGBoost identified over a hundred significant features per trait, indicating a more complex basis. The thesis concludes that the choice of algorithm and preprocessing technique depends on the research goal, with PCA-enabled XGBoost ideal for computational efficiency, and CatBoost on original data superior for generating specific and interpretable biological hypotheses.

Acknowledgements

Special thanks to:
Rosario Garcia Gil (SLU), Laura Morales (SLU), Sonali Ranade (SLU), Haleh Hayatgheibi (SLU) and Gustav Pihlgren(UMU) for guidance and feedback.

Contents

1 Introduction

The studies of genetics are among the most impactful studies on all forms of life, from humans, to animals, to plants and even bacteria and viruses. Through the discovery of thousands of genetic markers, researchers are able to understand the development of diseases and their mechanisms and develop personalized medicine and targeted therapeutic approaches. To this day, researchers are constantly searching for new methods that make it possible to decipher the genetic makeup of living organisms in order to understand their complex genomic interactions and how they affect the physical characteristics of these organisms. However, despite the effort, our understanding of genetics still is extremely limited.

Norway spruce (Picea abies) is a coniferous tree species native to Europe that grows in the Boreal forests of northern Europe and some parts of central and southern Europe, which has an economical and ecological importance, especially in Scandinavia, thanks to its desirable properties such as straightness and uniformity [12]. Moreover, it plays a vital role in biodiversity and soil and water conservation, as it provides habitats for various flora and fauna, stabilizes soil and contributes to water regulation, helping reduce erosion and maintain water quality in forested landscapes. Therefore, it is of great importance to study these species and understand their genetic makeup [3]. Genome-wide association studies (GWAS) can identify single nucleotide polymorphisms (SNPs) that contribute to phenotypic (physical) variations by analysing the genomes of sizeable populations and conducting experiments, but these experiments are conducted using traditional biological methods that are costly and time consuming. In the era of big data and artificial intelligence, supervised machine learning (ML) has emerged as an excellent technology for analysing and accurately predicting these intricate relationships [22]. Since genomic datasets are typically enormous and have high dimensions, ML is used in order to improve the efficiency of traditional methods and minimise their costs and time consumption. For this task, **XGBoost** and **CatBoost** regressors have been implemented and tested on a dataset containing the genomic sequences of 773 tree in order to find genotype-phenotype relationships. The phenotypic traits that are examined are Cellulose, Hemicellulose and Lignin.

1.1 Purpose and Research question

This thesis aims to evaluate and compare the performance of **Extreme Gradient Boosting (XGBoost) regressor** and **CatBoost regressor** when applied to high-dimensional genomic data to identify relationships and hidden pattrens in the data for Norway Spruce trees. In particular, it examines each model and the working principals behind it and investigates how **Principal Component Analysis (PCA)** for dimensionality reduction affects its prediction accuracy, computational efficiency, and training time. The research question of this thesis is:

- How does dimensionality reduction influence the performance and interpretability of XGBoost and CatBoost when trained on high-dimensional genomic data?

1.2 Thesis outline

In Chapter 2, the theoretical knowledge needed to understand the methodology and results of this thesis is provided, along with an introduction of the tools and models used in the experiment and their working principles. Chapter 3 discusses a few related studies in this field. The fourth Chapter describes the experiment and the method used to answer the research question. In Chapter 5, the results of the experiment are presented. Chapter 6 discusses and analyzes the outcomes of the experiment, and Chapter 7 presents the conclusion of this thesis and its limitations.

2 Theoretical background

2.1 Genome-wide association studies

GWAS are studies that focus on plants and agriculture with the aim to identify specific SNPs responsible for phenotypic traits and variations in several distinct populations [3]. A DNA block consists of several chains, and each chain consists of several stacked nucleotides. Each nucleotide contains one base (A, T, C, or G). A pair of nucleotide bases is often referred to as a genotype, which could be either homozygous (e.g., AA/GG) or heterozygous (e.g., AG). An SNP represents a genetic variation in an individual that occurs on one of the two pairs of zygotes, where one of the bases is substituted for another (e.g., a C $\rightarrow$ T change) on a specific position in the genome as Figure 1 shows, causing a genetic mutation in that individual, or a genetic variation in a group of individuals, if, and only if that variation occurs in at least 1% of the population [15].

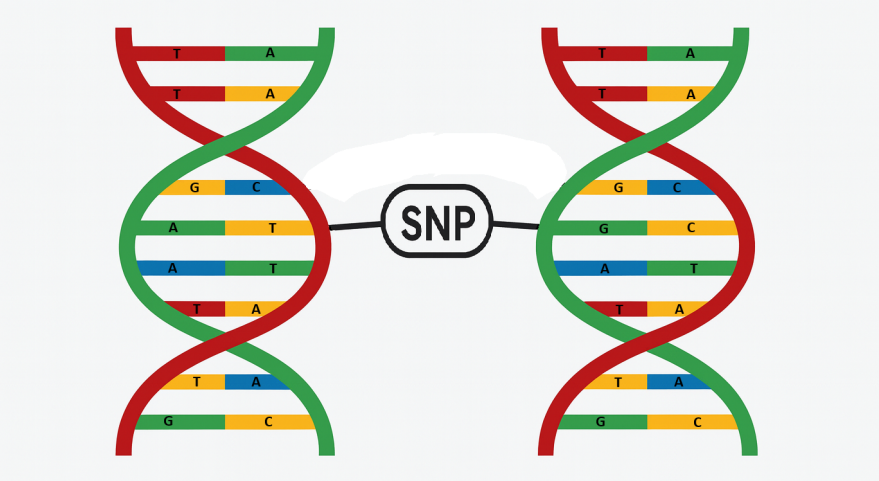


Figure 1: Single nucleotide polymorphism.

2.2 Regression

Regression is a subset of supervised machine learning used for predicting continuous outcomes. Regression models operate by modelling the relationship between a dependent variable (target) and one or more independent variables (predictors) [17]. The goal is to model the relationship between the predictors and the target variable. Through the training process, regression models learn from labelled data, adjusting their parameters to minimize the difference between the actual and predicted values, typically measured by errors such as Root Mean Square Error (RMSE) [10].

There are several types of regression models, each with its use cases, characteristics, weaknesses, and strengths. Among these models, Linear Regression, Decision Tree Regressors, and KNN are widely used for their distinct approaches to modelling and predicting data. Regres-

sion models often utilize hyperparameters in order to tune the models [4]. Parameters and hyperparameters are distinct elements that are crucial to the development of the models, with the major difference being that parameters are the configuration variables that are internal to the model which are learned from the data during training, and are essential for the models' prediction capabilities, while hyperparameters are the configuration settings or external configurations for the models and the learning process, which are not learned from the data but are set prior to the training process and remain constant during training, moreover, if the hyperparameters are not adjusted, the models can be prone to overfitting or underfitting.

2.3 Data Normalization

Data normalization is the process of scaling the data so that it falls in the range between zero and one [13], and this is done by dividing the difference between a value in a column and the minimum value in that column, by the difference between the maximum- and the minimum value in the column using the following function:

$$X_{norm} = \frac{X - min(X)}{max(X) - min(X)}$$

2.4 Decision Trees

Decision trees are a class of intuitive, non-parametric supervised learning algorithms used for both classification and regression tasks. Their structure resembles a tree of decisions and their possible consequences, making them highly interpretable [10, 4].

A decision tree is built through a process known as recursive partitioning. The algorithm starts at a root node containing the entire dataset and iteratively splits the data into subsets based on the value of a feature that best separates the data according to a specific metric such as variance reduction for regression tasks. This process continues, creating internal decision nodes and terminal leaf nodes, until a stopping criterion is met which could be a maximum tree depth or a minimum number of samples per node.

A primary advantage of decision trees is their simplicity and transparency, as the resulting model can be easily visualized and understood. However, they have a tendency to overfit the training data, creating complex trees that capture noise rather than the underlying pattern. Therefore, to overcome this limitation, decision trees are rarely used alone in modern machine learning. Instead, they serve as the fundamental weak learners (base estimators) in ensemble methods such as Random Forests and Gradient Boosting Machines [7, 20]. These ensemble methods utilize the predictions of many individual trees, reducing variance and significantly improving predictive accuracy and robustness.

2.5 Model Ensembles

Model ensembling is a powerful technique in machine learning that combines the predictions from multiple base models to produce a single, superior prediction. The core principle is that a collection of weak learners can form a strong learner, often achieving better accuracy, robustness, and generalizability than any single model could alone [4]. Common ensemble

techniques include Bagging and Boosting.

Bagging is a method that reduces variance and helps avoid overfitting by training multiple instances of the same base algorithm (e.g., decision trees) on different random subsets of the training data, drawn with replacement. For regression, the final prediction is typically the average of all individual models. A prominent example is the Random Forest algorithm.

Boosting is a sequential method that focuses on reducing bias. It trains models consecutively, where each new model is specifically tuned to correct the errors made by the previous ones. By assigning greater weight to misclassified instances, the algorithm iteratively improves performance on the most challenging parts of the dataset. Modern high-performance algorithms like XGBoost, CatBoost, and AdaBoost are all advanced implementations of gradient boosting, which use decision trees as their weak learners. These have become dominant in fields like bioinformatics and genomics for predicting complex traits [22].

2.6 Gradient Boosting

Gradient boosting is a machine learning technique belonging to the ensemble learning family, which combines multiple weak predictive models to create a single strong model. Boosting works by constructing models sequentially, with each new model attempting to correct the errors made by the previous ones [4]. The core principle of gradient boosting is to optimize an arbitrary differentiable loss function using gradient descent. The algorithm typically utilizes shallow decision trees as the weak learners (base models). The process iteratively fits new trees to minimize the residual errors of the current ensemble. This sequential refinement allows the model to capture complex patterns and relationships in the data that simpler models might miss.

2.7 XGBoost

XGBoost is a scalable and flexible machine learning algorithm based on gradient-boosted decision trees, which is favoured for its computational efficiency and predictive performance. It has become one of the most extensively used models in data science and scientific applications due to its ability to handle complex, high-dimensional data while minimizing overfitting [9]. It utilizes additive training and adds trees iteratively to correct residuals. Moreover, it uses weighted quantile sketches for efficient candidate splitting while automatically handling missing values during training. XGBoost optimizes a regularized objective function combining a differentiable loss function L and a penalty term Ω:

$$\text{Obj}(\theta) = \underbrace{\sum_{i=1}^{n} L(y_i, \hat{y}_i)}_{\text{Loss term}} + \underbrace{\sum_{k=1}^{K} \Omega(f_k)}_{\text{Regularization term}} \tag{2.1}$$

where:

- y_i is the true label for observation i
- $\hat{y}_i$ is the predicted label
- f_k represents the k-th tree in the ensemble
- $\Omega(f_k)$ is the regularization term for the k-th tree

The regularization term for each tree is defined as:

$$\Omega(f_k) = \gamma T + \frac{1}{2}\lambda \|w\|^2 \tag{2.2}$$

where:

- T is the number of leaves in the tree
- w is the vector of leaf weights (output values)
- γ is the minimum loss reduction required to split a node
- λ is the L2 regularization hyperparameter

2.8 CatBoost

CatBoost is a gradient boosting algorithm developed by Yandex [20], specifically designed to handle datasets with categorical features. Unlike traditional boosting methods, CatBoost efficiently processes categorical variables without extensive preprocessing, such as one-hot encoding, and avoids overfitting through ordered boosting and permutation-driven techniques. CatBoost builds an ensemble of decision trees sequentially, where each tree attempts to minimize a specific loss function. The training is based on gradient boosting, where new trees are added to reduce the residual error of the ensemble [20]. CatBoost is formalized as follow:

Let the training dataset be $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{n}$, where $x_i \in \mathbb{R}^d$ represents the feature vectors and $y_i \in \mathbb{R}$ denotes the target values. CatBoost builds an additive model in a forward stagewise manner:

$$F(x) = \sum_{t=1}^{T} \eta_t h_t(x),$$

where:

- $F(x)$ is the final prediction function,
- T is the number of boosting iterations (trees),
- $h_t(x)$ is the prediction of the t^{th} tree,
- $\eta_t \in (0, 1]$ is the learning rate at iteration t.

At each iteration, a new tree $h_t(x)$ is trained to minimize the gradient of the loss function L with respect to the ensemble prediction from the previous iteration $F_{t-1}(x)$:

$$h_t(x) = \arg\min_{h} \sum_{i=1}^{n} \left[\left(\frac{\partial L(y_i, F_{t-1}(x_i))}{\partial F_{t-1}(x_i)} \right) h(x_i) \right].$$

2.9 Shapley Additive Explanations

As machine learning models, particularly complex ensemble models like gradient boosting machines and deep neural networks, become more powerful and prevalent, the need to understand their predictions has become critical.

SHAP is a tool for explaining the output of any machine learning model by calculating the contribution of each feature to a specific prediction. It works by assigning every feature an importance value (a SHAP value) that represents how much that feature moved the model's prediction away from the baseline average [14]. This approach provides both local interpretability, explaining individual predictions by highlighting the impact of each input variable, and global interpretability, which can be achieved by examining these values across many instances to reveal overall model behaviour and consistent, accurate feature importance.

In the context of a GWAS for Norway spruce, SHAP could be used to explain a predictive model for traits like cellulose content. It would not just identify which SNPs are important, but would quantify and illustrate how each SNP's genotype (e.g., homozygous vs. heterozygous) contributes to the final prediction for an individual tree, making it easier to interpret complex models and understand their behaviour.

2.10 Principal Component Analysis

Principal Component Analysis (PCA) is a widely used unsupervised linear transformation technique for dimensionality reduction that projects high-dimensional data into a lower-dimensional subspace while preserving maximal variance. By computing the eigenvectors of the covariance matrix, PCA identifies orthogonal principal components (PCs) that capture the dominant patterns in the data, ordered by their explained variance, which refers to how much of the total variability in the dataset is captured by each PC [11]. This method is particularly effective for noise reduction, feature extraction, and visualization of complex datasets, such as genomic or image data [18]. Given a centered data matrix $\mathbf{X} \in \mathbb{R}^{n \times p}$ with n samples and p features, PCA performs:

1. **Covariance Matrix Computation**:
$$\mathbf{C} = \frac{1}{n-1}\mathbf{X}^\top \mathbf{X} \tag{2.3}$$
2. **Eigenvalue Decomposition**:
$$\mathbf{C} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^\top \tag{2.4}$$
where $\mathbf{V}$ contains eigenvectors (principal directions) and $\mathbf{\Lambda}$ is a diagonal matrix of eigenvalues.
3. **Dimensionality Reduction**: Project data onto top-k principal components:
$$\mathbf{Z} = \mathbf{X}\mathbf{V}_{:k} \tag{2.5}$$
where $\mathbf{V}_{:k}$ contains the first k eigenvectors.

The proportion of variance explained by the j-th PC is:
$$\frac{\lambda_j}{\sum_{i=1}^{p} \lambda_i} \tag{2.6}$$

2.11 Evaluation metrics

This work uses Root Mean Square Error (RMSE) as the evaluation metric. The RMSE is a widely used metric to evaluate the accuracy of regression models by measuring the difference

between predicted and observed values. It computes the square root of the average squared errors, providing a measure of prediction error in the same units as the target variable [6]. Lower RMSE values indicate better model performance [6]. The RMSE is calculated according to the following formula:

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

Predicted values: $\hat{y}_1, \hat{y}_2 \ldots . \hat{y}_n$. Actual values: $y_1, y_2 \ldots y_n$

2.12 Cross-Validation

Cross-validation (CV) is a fundamental resampling technique used to assess the generalizability of a predictive model and to minimize the risk of overfitting. The core principle involves partitioning the available data into smaller subsets, training the model on one subset (the training set), and validating it on the other subset (the validation or test set) that was not used during training, which in turn, can help give an accurate estimate of its expected performance on unseen data [8]. The most common variant is k-fold cross-validation. In this approach, the dataset is randomly shuffled and split into k equal-sized, independent folds. The model is then trained k times, in each iteration, k-1 folds are used for training, and the remaining single fold is held back for validation. This process is repeated until each fold has served as the validation set exactly once. The final performance metric, such as RMSE, is calculated as the average of the results from all k iterations [21].

2.13 Overfitting and Underfitting

Overfitting occurs when a regression model learns the training data too well, including its noise and outliers, rather than just the underlying trend or pattern. While underfitting occurs when a regression model is too simple to capture the underlying structure of the data. This happens when the model does not have enough parameters [4].

3 Related works

The authors in [16] conducted a comparative study of machine learning models for predicting crop yields of Bengal gram, groundnut, and maize in Telangana, India, using data from 2016 to 2018. The study evaluated six regression algorithms: Linear Regression, Decision Tree Regression, Gradient Boosting Regression, Random Forest Regression, XGBoost Regression, and an ensemble Voting Regressor. After extensive preprocessing and hyperparameter tuning, the performance of each model was assessed using Mean Absolute Error (MAE), Mean Squared Error (MSE), R^2 score, and K-fold Cross-Validation. Among all models, Random Forest Regression demonstrated the best performance, achieving an average R^2 score of 0.981, with significantly lower MAE and MSE values compared to the other approaches. The study concludes that Random Forest is a robust and efficient model for yield prediction in the agricultural domain, offering both high accuracy and generalizability. This work demonstrates the potential of machine learning to enhance agricultural decision-making and resource optimization through reliable crop yield forecasting.

In [1], a comparative analysis using the machine learning models CatBoost, XGBoost, Random Forest, and Support Vector Machines was conducted to predict breast cancer status from high-dimensional gene expression data derived from the TCGA-BRCA dataset. The dataset contained over 20,000 gene expression features per sample, presenting a typical high-dimensional challenge. Dimensionality reduction was performed using feature selection techniques based on mutual information, followed by model training and evaluation using 10-fold cross-validation. Performance metrics included Accuracy, Precision, Recall, F1-score, and AUC-ROC. CatBoost outperformed all other models in terms of AUC-ROC (0.985) and F1-score (0.964), while maintaining relatively stable training times and requiring fewer hyperparameter adjustments due to its built-in support for categorical features and regularization. The study concluded that CatBoost is a highly effective tool for biomedical prediction tasks involving high-dimensional omics data.

The authors in [13] developed an ensemble machine learning model called Ada-XG-CatBoost, which combines XGBoost and CatBoost regressors within an AdaBoost framework, enhanced through Bayesian hyperparameter optimization. The model was applied to predict Gross Ecosystem Product (GEP), an ecological economic indicator, using a high-dimensional environmental dataset collected from Chinese national city-level statistics. Dimensionality reduction and feature selection were performed during preprocessing, and the model's performance was benchmarked against standalone XGBoost and CatBoost using R^2, MAE, and RMSE metrics. The ensemble model achieved superior results with $R^2 = 0.9992$, MAE = 0.0093, and RMSE = 0.0089, clearly outperforming XGBoost ($R^2 = 0.9976$) and CatBoost ($R^2 = 0.9973$). Furthermore, SHAP analysis highlighted regional evaporation, average precipitation, and NDVI as the most influential features. These findings show that the Ada-XG-CatBoost model offers improved generalization, robustness, and interpretability for complex ecological forecasting tasks involving high-dimensional data.

In [5], the authors conducted a machine learning analysis of Alzheimer's disease genetics

using genome-wide data from 41,686 individuals. Multiple algorithms were tested, including Gradient Boosting Machines, Neural Networks, and MB-MDR, with performance assessed in terms of their ability to replicate known loci, identify novel variants, and predict disease risk. Gradient boosting models demonstrated strong predictive accuracy and were able to uncover genetic associations beyond those detected by traditional GWAS methods. The study concludes that boosting approaches are particularly effective for high-dimensional genomic datasets, providing both accurate prediction and the potential for novel biological discovery.

In [2], the authors developed a machine learning–based feature selection framework (ML-FS) to enhance prediction accuracy of coronary artery disease (CAD) using SNP data from the UK Biobank. The study compared the ML-FS approach, which reduced the dataset to a panel of around 50 informative SNPs, against traditional polygenic risk score (PRS) methods. Several machine learning models were applied and evaluated using metrics such as prediction accuracy and robustness. The ML-FS method achieved superior results, reaching accuracies of nearly 80% while using far fewer variants than PRS. The authors conclude that integrating machine learning with targeted feature reduction provides an efficient and reliable approach for genomic prediction tasks.

In [19], the authors explored the application of deep learning techniques to genome-wide association data for predicting pathogenic phenotypes in Group A Streptococcus (GAS). The dataset consisted of high-dimensional genomic variants derived from GWAS experiments, posing a similar challenge to other omics studies in terms of dimensionality and noise. To address this, the researchers implemented and compared several neural network architectures, including convolutional neural networks (CNNs), residual networks (ResNets), and recurrent long short-term memory (LSTM) models. Each architecture was trained and evaluated under two conditions: using the complete set of SNP features and using dimensionally reduced feature sets. The results revealed that networks trained on the full SNP datasets consistently achieved higher predictive performance, while those trained on reduced datasets showed a notable decline. However, certain architectures, such as ResNets, demonstrated partial robustness to dimensionality reduction. The authors conclude that while dimensionality reduction offers computational advantages, it risks discarding biologically relevant signals.

While the above studies demonstrate the potential of machine learning for high-dimensional biological and ecological data, this thesis extends the previous research in two important directions. First, unlike crop yield forecasting [16], breast cancer prediction [1], ecological indicators [13], or Alzheimer's disease genetics [5], this thesis applies gradient boosting methods directly to Norway spruce genomic data, which is both underexplored and ecologically significant. Second, by systematically comparing XGBoost and CatBoost under both original and PCA-reduced feature spaces, the thesis contributes a detailed analysis of how dimensionality reduction impacts not only predictive accuracy and computational cost, but also model interpretability via SHAP. This focus on the trade-off between efficiency and biological insight bridges the gap between purely predictive frameworks [19] and feature-selection-based genomic analyses [2], embedding this experiment within existing approaches while highlighting its novelty in the forestry and plant genomics domain.

4 Method

This chapter brings forth the methodological framework designed to investigate the influence of dimensionality reduction using PCA on the performance of XGBoost and CatBoost regressors for genomic prediction.

4.1 Hardware and software measurements for a fair comparison

To ensure a fair result, multiple factors have been taken into consideration before proceeding with the experiment such as hardware and software consistency, and for that purpose, the following measurements are applied to avoid environmental deviations throughout all stages of testing, training, and validation for the different regression models:

- Hardware and Software Consistency: By using one computer, we can ensure that all models are evaluated on the same hardware and software. This includes the same CPU, GPU, machine learning libraries and dependencies etc. Differences in hardware or software can significantly impact the computation time, model training speed, and even the final performance metrics of the models.
- Random Seed Control: It allows for better control over random processes, such as data shuffling and initialization of model weights. Setting the same random seed across all model trainings ensures that any stochastic elements in the training process do not introduce bias in the comparison.

4.2 Dataset

The dataset used in this thesis was assembled by researchers at Sveriges Lantrbruksuniversitet (SLU). SLU is a Swedish public university with a core focus on the sustainable use of biological natural resources. The assembled data was extracted from 773 tree samples of Norway Spruce gathered from Sävar in northern Sweden.

The original dataset contained the genotypic, and phenotypic information of the samples for this task, and it was delivered in three files. The first file containing the genotypic data of the samples which was filtered for biallelic values where minor allele frequency is > 1%, and samples with 10% < missingness were filtered out as well. The data was ordered as in Table 1 where:

- CHROM (Chromosome) is the identifier of the chromosome on which the genetic variant is located.
- POS (Position) is the precise physical position (base pair number) of the genetic variant on the chromosome.

- REF (Reference Allele) is the nucleotide (A, C, G, or T) that is found at this position in the reference genome for the species.
- ALT (Alternate Allele) is the nucleotide that varies from the reference genome, i.e., the alternative variant or "minor" allele at this position.
- TREE1, TREE2, ... TREEn is the Genotype Calls. These columns contain the genotypic data for each individual tree at this specific location. The format 0/0, 0/1, 1/1 is standard VCF notation where 0 and 1 correspond to the reference allele and alternative allele respectively.

The second containing the phenotypic data of the samples ordered as in Table 2 where:

- TREE is the tree identifier, which is unique ID for each individual tree sample. This links the phenotypic data to the genotypic data in Table 1 and the location data in Table 3.
- Cellulose is the percentage of cellulose content in the wood of the tree.
- Hemicellulose is the percentage of hemicellulose content in the wood.
- Lignin is the percentage of lignin content in the wood.

And the third containing information regarding the precise geographical location, from which the samples were taken, along with the data produced by running a population structure assessment that was conducted at SLU as shown in Table 3.

- TREE is the tree identifier.
- Lat (Latitude) is the geographical latitude from which the tree sample was originally taken.
- Lon (Longitude) is the geographical longitude from which the tree sample was originally taken.
- PCA1, PCA2, ... PCAn (Principal Components). These columns contain the values for each tree from a PCA performed on the genetic data to captures patterns of genetic relatedness and population structure within the sample set.

Table 1 The genotypic data of the samples.

CHROM	POS	REF	ALT	$TREE_1$	$TREE_2$	...	$TREE_n$
MA_10001558	1963	C	T	0/0	0/1	...	1/0
MA_10001554	6727	A	T	0/1	1/1	...	0/0
MA_10001552	3435	A	G	1/1	0/1	...	1/1

Table 2 The phenotypic data of the samples.

TREE	Cellulose	Hemicellulose	Lignin
$TREE_1$	22.343	44.223	22.432
$TREE_2$	33.231	23.345	33.253
$TREE_3$	24.457	41.372	47.342

Table 3 The coordinates & population structure assessment of the samples.

TREE	Lat.	Lon.	PCA_1	...	PCA_n
$TREE_1$	64.95	18.58	0.005305864	...	0.013747994
$TREE_2$	64.58	18.45	0.002888573	...	0.001130156
$TREE_3$	60.02	17.92	0.009974156	...	0.001276022

4.3 Data management

4.3.1 Data preprocessing

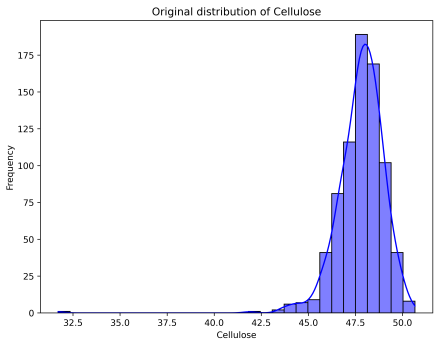
To understand the task at hand and what is being investigated, it is essential to take a deeper look at the data and process it in a way that maximizes the models' performance. The first step was to encode the data using One-Hot Encoding as follows:

- 1 for the first homozygote type (AA)
- 2 for the other homozygote type (BB)
- 3 for the heterozygote types (AB/BA)

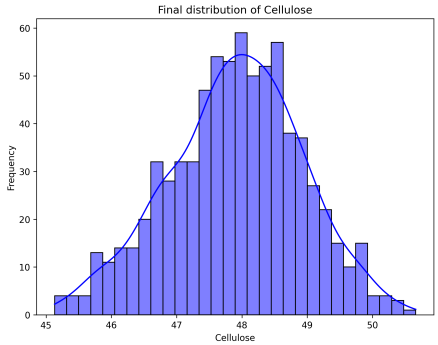
To minimize the dataset's dimensionality while preserving variance, feature engineering had to be performed by merging the CHROM and POS columns into a composite CHROM_POS identifier, and pivoting the data to restructure it into a genomic sequence matrix, where each row represented a unique tree and each column corresponded to a specific SNP locus. The final step was to concatenate the population structure data to the genotypic data. This transformation compressed the dataset to 226,640 dimensions while maintaining critical genetic variation as shown in Table 4.

4.3.2 Data preparation

Before feeding the data to the models, an analysis of the phenotypic data was conducted in order to remove outliers and transform the values into a form that is suitable for the models. For Cellulose, Hemicellulose, and Lignin, respectively 15, 16 and 18 outliers were removed due to the skewness of the data. Figures 2, 3, and 4 illustrate the original and the final distribution of each target variable.

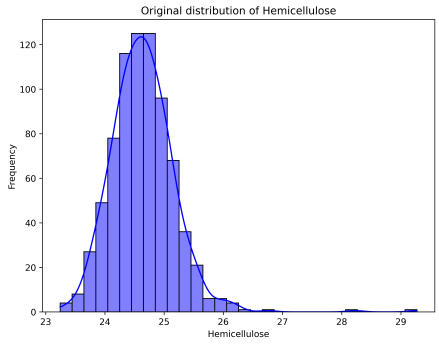


(a): Original distribution of Cellulose

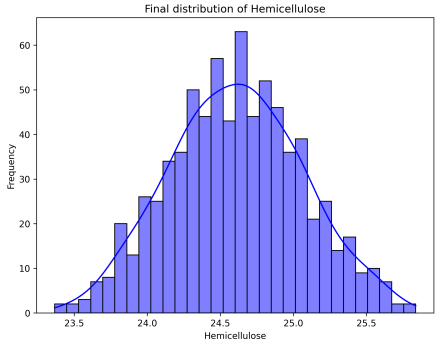


(b): Final distribution of Cellulose

Figure 2: Distribution of Cellulose

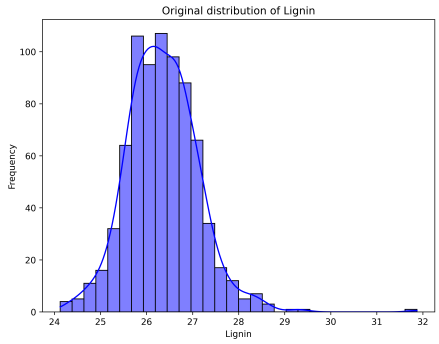


(a): Original distribution of Hemicellulose

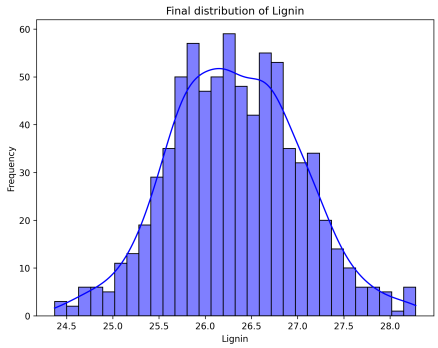


(b): Final distribution of Hemicellulose

Figure 3: Distribution of Hemicellulose



(a): Original distribution of Lignin



(b): Final distribution of Lignin

Figure 4: Distribution of Lignin

Table 4 The final dataset.

TREE	$CHROM_1\_POS_1$	...	$CHROM_n\_POS_n$	Lat.	Lon.	PCA_1	...	PCA_n
$TREE_1$	1	...	3	64.95	18.58	0.00604121	...	0.013747994
...	...	...	...	...	...	...	...	...
$TREE_n$	2	...	3	64.95	18.58	0.00604881	...	0.013747937

4.4 Dimensionality reduction

PCA has been applied on the dataset to reduce its dimensions as much as possible without losing much variance. This was done to create two new datasets, once by setting the variance threshold to 85%, and once by setting it to 95%. The results of running PCA showed that 589 components accounted for 85% variance, and 696 components accounted for 95% variance as shown in Figure 5 .

4.5 Implementation and training

The implementation and training of the models were designed to robustly handle the high-dimensional genomic data while optimizing hyperparameters to maximize predictive per-

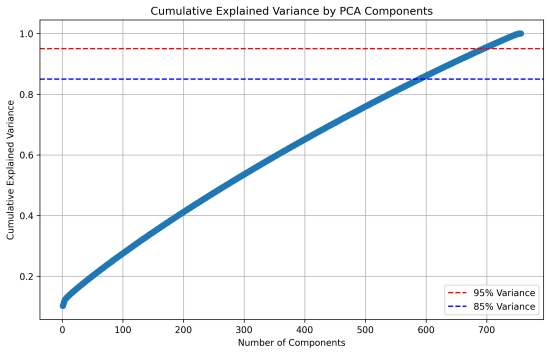


Figure 5: Cumulative explained variance by PCA components

formance for the target traits cellulose, hemicellulose, and lignin content. The process was implemented in Python, using key libraries including xgboost, catboost, scikit-learn, numpy, pandas and more. The training pipeline consisted of several stages as follows:

1. Data Partitioning:
   The feature matrix X (containing genotypic and population structure data) and target vector y (containing the phenotypic trait values) were first split into training and testing sets using an 80/20 ratio using the train_test_split function from scikit-learn. A fixed random seed (random_state=42) was set at this stage to ensure the reproducibility of all subsequent results.

2. Hyperparameter Grid:
   A grid of hyperparameters was defined for the search, containing key parameters that control model complexity. Table 5 and Table 6 detail the hyperparameters that were tested along with their search spaces for both XGBoost and CatBoost respectively. The combination that yielded the lowest average validation RMSE across all folds was selected as the optimal configuration.

Table 5 Hyperparameter search space for the XGBoost model.

Hyperparameter	Description	Search Range
n_estimators	The number of boosting rounds (trees) to build.	[50, 400]
max_depth	The maximum depth of a tree. Increases model complexity.	[2, 7]
learning_rate	Step size shrinkage used to prevent overfitting.	[0.01, 0.3]
subsample	Fraction of instances (rows) to be randomly sampled for each tree.	[0.2, 1.0]
colsample_bytree	Fraction of features (columns) to be randomly sampled for each tree.	[0.2, 1.0]
min_child_weight	Minimum sum of instance weight (Hessian) needed in a child node.	[1, 7]
reg_alpha	L1 regularization term on weights.	[0.1, 1.0]
reg_lambda	L2 regularization term on weights.	[0.1, 1.0]

3. Custom Cross-Validation and Hyperparameter Tuning:
   A central component of the method was implementing a custom 5-fold CV function

Table 6 Hyperparameter search space for the CatBoost model.

Hyperparameter	Description	Search Range
iterations	The maximum number of trees that can be built.	[100, 500]
depth	The maximum depth of each tree.	[3, 7]
learning_rate	The step size shrinkage used to prevent overfitting.	[0.01, 0.3]
l2_leaf_reg	Coefficient for L2 regularization on leaf weights.	[1.0, 10.0]
subsample	The fraction of the training data to use for each tree.	[0.6, 1.0]
min_data_in_leaf	The minimum number of training samples in a leaf.	[1, 100]
random_strength	The amount of randomness to use for scoring splits.	[0.1, 1.0]

with manual grid search. This approach was chosen over automated methods to maintain fine control over the data flow, especially concerning the normalization of the target variable within each fold to prevent data leakage. This function was executed 100 times during each training instance, utilizing a different combination of hyperparameters each time. The function included the following:

- K-Fold Splitting: For each hyperparameter combination in the grid, the training set (X_train, y_train) was further split into five folds using K-Fold from scikit-learn. Four folds were used for training, and one fold was used for validation in each iteration.
- Target Normalization per Fold: To prevent information leakage from the training data to the validation data within the CV loop, a MaxAbsScaler was fit exclusively on the training portion of the fold. The scaler was then used to transform both the training and validation targets for that fold. This ensured that the scaling parameters were derived only from the training subset in each split.
- DMatrix Conversion for XGBoost: The scaled data for each fold was converted into XGBoost's optimized data structure, DMatrix, for efficient memory handling and training, especially with GPU acceleration enabled. This conversion was performed only for the XGBoost model.
- Early Stopping and Evaluation: The model was trained on the fold's training data with early stopping activated (early_stopping_rounds=10), using the validation set to monitor the RMSE. Training stopped if the validation error failed to improve for 10 consecutive rounds, and the model from the best iteration was retained.
- RMSE Calculation: Predictions were made on the validation fold. These predictions, along with the validation targets, were then inverse-transformed back to their original scale using the fold-specific scaler before the RMSE was calculated. The final performance metric for a hyperparameter set was the average RMSE across all five folds.

4. Final Model Training:
After identifying the best hyperparameters and the most frequently optimal number of boosting rounds from the custom CV function, the final model was trained on the

entire training set (X_train, y_train). The target variable for the full training set was normalized using a MaxAbsScaler fit on this entire set.

5. Final Evaluation:
   The final trained model was evaluated on the test set (X_test, y_test), which had not been used during any part of the tuning process. Predictions were generated, inverse-transformed to the original scale, and the final test RMSE was computed. This metric provides an unbiased estimate of the model's performance on unseen data.

6. SHAP Computation:
   A shap.TreeExplainer object was initialized using the trained model (final_model). The SHAP values were then calculated for the entire feature matrix (X) used in training. These values represent the contribution of each feature to the model's prediction for every individual sample in the dataset.

   To assess global feature importance, the mean absolute SHAP value was computed for each feature across all samples. This metric provides a robust measure of a feature's overall impact on the model's output, as it averages the span of contributions regardless of their positive or negative direction. The results, including the feature names and their corresponding mean absolute SHAP values, were stored in a DataFrame for further analysis.

   The resulting list of features, ranked by their mean absolute SHAP value, was used to perform feature elimination. Any feature with a mean absolute SHAP value of zero was deemed unimportant and was removed from the dataset. This criterion indicates that the feature had no contribution to any of the model's predictions. The process resulted in a new, refined dataset containing only the features with non-zero impact, further enhancing model interpretability by isolating the most genetically relevant markers for each target trait.

The preceding six steps formed the core of an iterative feature elimination loop, where insignificant features were removed in each iteration. The loop terminated when there were no insignificant features left to remove, resulting in a final dataset composed exclusively of features with significant contributions to the model's predictions for each target trait.

4.6 Evaluation of the models

The models were evaluated by calculating the RMSE and computational time for each target variable and each dataset. That includes the original datasets and the datasets transformed with PCA. This is done to calculate the effects of PCA on prediction accuracy and training time.

5 Results

This chapter presents the findings of the experiments conducted to evaluate the performance of the XGBoost and CatBoost regression models on the Norway spruce genomic dataset. The models were trained and evaluated on three distinct datasets: the original high-dimensional dataset, a dataset reduced via PCA to retain 95% of the variance, and a dataset reduced to retain 85% of the variance. The evaluation metrics are Root Mean Squared Error (RMSE) for predictive accuracy and computational time for efficiency. The results presented here were calculated using the trained models from the very first iteration of the experiment; in other words, the models trained on the datasets before any feature elimination was conducted.

5.1 XGBoost

The performance of the XGBoost regressor across the different dataset configurations is summarized in Table 7. The results indicate a notable trade-off between predictive accuracy and computational efficiency.

In terms of prediction error, XGBoost demonstrated robust performance across all datasets. For the prediction of cellulose content, the RMSE remained stable at 1.02 for both the original and the 95% variance datasets, with only a marginal increase to 1.03 on the 85% variance dataset. A similar pattern was observed for hemicellulose, where the RMSE was consistent at 0.44 for the original and 95% variance datasets, rising slightly to 0.45 for the 85% variance dataset. The model's performance for lignin content was also stable, with an RMSE of 0.71 on the original data and 0.72 on the 95% variance data, increasing to 0.74 on the 85% variance data.

The most significant impact of PCA was observed in the computational time required for training. The original dataset required extensive computational resources, with training times of 35 hours for cellulose, 35 hours for hemicellulose, and 40 hours for lignin. The application of PCA drastically reduced this time. Using the 95% variance dataset, training times were reduced to approximately 30 minutes for all traits. A further reduction to 85% variance compressed the training time to just 20-25 minutes, representing a reduction of over 99% compared to the original dataset without a substantial sacrifice in predictive accuracy.

Following the iterative feature elimination process, XGBoost identified a substantial number of SNPs as significant contributors to each trait. The model retained 112 features for cellulose, 149 features for hemicellulose, and 120 features for lignin, suggesting that these traits exhibit a relatively large and complex set of genetic markers.

Table 7 Evaluation results for XGBoost

	Cellulose		**Hemicellulose**		**Lignin**	
	RMSE	Comp. time	RMSE	Comp. time	RMSE	Comp. time
Original	1.02	35 Hours	0.44	35 Hours	0.71	40 Hours
95%	1.02	29 Minutes	0.44	30 Minutes	0.72	32 Minutes
85%	1.03	23 Minutes	0.45	22 Minutes	0.74	25 Minutes

5.2 CatBoost

The CatBoost regressor displayed strong predictive performance on the original dataset, but showed a greater sensitivity to dimensionality reduction compared to XGBoost, as detailed in Table 8.

On the original high-dimensional dataset, CatBoost achieved a marginally higher RMSE value on Hemicellulose with a score of 0.45, but lower RMSE values on Cellulose and Lignin with scores of 1.01 and 0.68, respectively. However, the application of PCA had a more pronounce effect on its accuracy. For cellulose, the RMSE increased to 1.04 and 1.06 on the 95% and 85% variance datasets, respectively. The RMSE for hemicellulose remained at 0.45 for the 95% variance dataset but increased to 0.47 for the 85% variance dataset. The most notable decline was for lignin, where the error increased from 0.68 on the original data to 0.70 on the 95% variance dataset and 0.75 on the 85% variance dataset

In terms of computational efficiency, PCA resulted in a massive advantage for CatBoost as well. Training times on the original dataset were considerable, ranging from 32 to 37 hours. This was reduced to an average of 27 minutes on the 95% variance dataset and further down to approximately 20 minutes on the 85% variance dataset. This represents a reduction in computational time of over 98%, making the training process vastly more efficient.

A key finding from the feature elimination process was the significant difference in model interpretability. CatBoost identified drastically fewer features as significant compared to XGBoost. It retained only 40 features for cellulose, and a very minimal set of 5 features for hemicellulose and 4 features for lignin. This suggests that CatBoost attributes the predictive power for hemicellulose and lignin to a very select few genomic markers.

Table 8 Evaluation results for CatBoost

	Cellulose		**Hemicellulose**		**Lignin**	
	RMSE	Comp. time	RMSE	Comp. time	RMSE	Comp. time
Original	1.01	32 Hours	0.45	33 Hours	0.68	37 Hours
95%	1.04	27 Minutes	0.45	26 Minutes	0.70	30 Minutes
85%	1.06	20 Minutes	0.47	18 Minutes	0.75	22 Minutes

6 Discussion and analysis

The results demonstrate a critical three-way trade-off between computational efficiency, predictive accuracy, and model interpretability, mediated by the choice of algorithm and preprocessing technique. While both models achieved strong performance on the original data, PCA-induced dimensionality reduction drastically reduced training times by over 98% for both algorithms, although with differing impacts on accuracy and interpretability. XGBoost exhibited remarkable robustness, maintaining stable RMSE across all dataset configurations, which suggests that the features discarded in the PCA process (aimed at capturing global variance) were largely redundant or noisy for its predictive task. However, this gain in speed and maintained accuracy comes at a cost to interpretability, by transforming the original features into composite principal components, PCA might struggle with the biological meaning of individual SNPs, making it difficult to trace a prediction back to a specific genetic variant.

In contrast, CatBoost's accuracy declined more noticeably on the reduced datasets, indicating a potential reliance on a more complex feature set that PCA accidentally removes. The most striking finding emerges from the SHAP-based feature elimination on the original data: CatBoost's identification of a minimal set of highly predictive features for hemicellulose 5 and lignin 4 suggests these traits may be led by a few key genomic markers, offering a highly interpretable model for biological insight. On the other hand, XGBoost's selection of over a hundred significant features for each trait points to a more complex architecture. This fundamental difference highlights the core interpretability trade-off: PCA enhances computational efficiency but diminishes genetic interpretability, while direct feature elimination on the original data preserves biological meaning at a higher computational cost.

6.1 Impact of PCA on interpretability

As PCA served as an effective tool for dimensionality reduction, its application fundamentally transformed the nature of the features used by the models, with profound implications for biological interpretation.

Principal Components are linear combinations of all original SNPs, engineered to capture maximum variance in the dataset. While this makes them statistically powerful for prediction, they are inherently abstract constructs that lack direct biological meaning. When XGBoost or CatBoost models are trained on PCA-transformed data, they learn to make predictions based on these composite components (e.g., PC1, PC2, ... PCn) rather than individual SNPs. Consequently, the output of the SHAP analysis identifies which principal components were most important for the model's prediction, not which specific genetic variants.

This presents a significant barrier to the primary goal of a GWAS: to identify candidate genes and SNPs associated with a trait. To trace a model's prediction back to an original SNP, extra measures are necessary to decompose the important PCs. This process involves examining the loadings (weights) of each original feature within the significant principal components. A SNP with a high absolute loading value on an important PC is itself likely to be influential. However, this decomposition is non-trivial and introduces additional complexity:

- Loss of Directness: The direct, clear link between a specific SNP genotype and a predicted phenotypic value, as provided by SHAP on the original data, is lost. Interpretation becomes a two-stage process: first identify important PCs, then investigate which SNPs contribute heavily to those PCs.
- Ambiguity in Decomposition: A single PC often has high loadings for dozens or even hundreds of SNPs. Understanding which of these are truly biologically related from those that are merely correlated in the population structure is challenging.
- Dilution of Effect: The effect of a single causative SNP may be diluted within a PC that is primarily driven by other, non-causal variants that happen to share a similar pattern of variation.

This explains why the feature elimination process was only biologically meaningful when applied to models trained on the original dataset. In that scenario, SHAP values could be directly computed for each SNP, providing a clear, quantifiable measure of its contribution, and allowing for the elimination of features with zero impact. In contrast, applying a similar elimination process to PCA components would merely remove composite features, offering no insight into the underlying genetic architecture.

Therefore, the use of PCA, while beneficial for computational performance, sacrifices the intuitive interpretability that is crucial for generating testable biological hypotheses. It trades the ability to pinpoint specific SNPs for the ability to train a model quickly on a compressed representation of the data. For research where the goal is pure prediction, this is an acceptable trade-off. However, for studies aimed at discovery and understanding genetic mechanisms, models trained on the original features, despite their computational cost, remain crucial for extracting meaningful insights about important SNPs.

6.2 Limitations

This thesis acknowledges certain limitations. The analysis was conducted on a specific dataset of Norway spruce from a single geographic region, which may limit the generalizability of the findings. Furthermore, the feature elimination process relies on the models' inherent biases. Finally, the biological validation of the identified candidate SNPs remains as critical future work to confirm their role in the phenotypic expression of the studied traits.

7 Conclusion

The findings of this thesis reveal that the optimal approach depends on the primary research goal. The primary conclusion is that PCA is an exceptionally effective tool for minimizing the computational burden of training on genomic data, reducing training times by over 98%. This comes with a variable cost: XGBoost proved to be highly robust to dimensionality reduction, with only a marginal increase in RMSE, making it an ideal choice when computational resources are limited and pure predictive power is the priority. CatBoost, while achieving the best absolute performance on the original data, was more sensitive to PCA.

This work highlights the profound influence of PCA on interpretability. PCA trades biological interpretability for computational efficiency by creating composite features that are statistically powerful but with a potential of ignoring important genetic relationships. Therefore, if the research objective is to generate specific, testable biological hypotheses about key genetic drivers, then models trained on the original data with subsequent feature elimination are essential. CatBoost excelled in this regard, pinpointing a very limited set of candidate SNPs.

Bibliography

[1] Rana Abdu-aljabar, Khansaa Aljafaar, Zinah Mohammed Ameen, and Hala Abduladheem. A comparative study of breast cancer detection and recurrence prediction using catboost classifier. *Acta Polytechnica*, 65:136–142, 05 2025.

[2] Z. Alireza, M. Maleeha, M. Kaikkonen, and V. Fortino. Enhancing prediction accuracy of coronary artery disease via ml-feature selection (ml-fs). *Journal of Translational Medicine*, 22(1):5090, 2024.

[3] John Baison, Amaryllis Vidalis, Linghua Zhou, Zhi-Qiang Chen, Zitong Li, Mikko J. Sillanpää, Carolina Bernhardsson, Douglas Scofield, Nils Forsberg, Thomas Grahn, Lars Olsson, Bo Karlsson, Harry Wu, Pär K. Ingvarsson, Sven-Olof Lundqvist, Totte Niittylä, and M Rosario García-Gil. Genome-wide association study identified novel candidate loci affecting wood formation in norway spruce. *The Plant Journal*, 100(1):83–100, 2019.

[4] Benjamin Johnston Blaine Bateman, Ashish Ranjan Jha and Ishita Mathur. *The the Supervised Learning Workshop : A New, Interactive Approach to Understanding Supervised Learning Algorithms*. Packt Publishing, 2020.

[5] Matthew Bracher-Smith, Alice Smith, and colleagues. Machine learning in alzheimer's disease genetics. *Nature Communications*, 16(1):1234, 2025.

[6] Tianfeng Chai and Roland R Draxler. Root mean square error (rmse) or mean absolute error (mae)? *Geoscientific Model Development*, 7(1):1247–1250, 2014.

[7] Tianqi Chen and Carlos Guestrin. XGBoost: A Scalable Tree Boosting System. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '16)*, page 785:794. ACM, 2016.

[8] Sean W. Fleming and David C. Garen. Simplified cross-validation in principal component regression (pcr) and pcr-like machine learning for water supply forecasting. *JAWRA Journal of the American Water Resources Association*, 58(4):517–524, 2022.

[9] Tong He and Michaël Benesty. Xgboost: A scalable tree boosting system. In *The Springer Series in Challenges in Machine Learning*, pages 1–37. Springer International Publishing, 2021.

[10] Benjamin Johnston and Ishita Mathur. *Applied supervised learning with Python : use scikit-learn to build predictive models from real-world datasets and prepare yourself for the future of machine learning*. International series of monographs on physics. Birmingham : Packt Publishing Ltd, 2019.

[11] I. T. Jolliffe and J. Cadima. Principal component analysis: A review and recent developments. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 374(2065):2–4, 2016.

[12] Juho Korhonen and Heikki Kujala. Economic and ecological importance of norway spruce (picea abies) in northern europe: A review. *Annals of Forest Science*, 76(3), 2019.

[13] Yang Liu, Tianxing Yang, Liwei Tian, Bincheng Huang, Jiaming Yang, and Zihan Zeng. Ada-xg-catboost: A combined forecasting model for gross ecosystem product (gep) prediction. *Sustainability*, 16(16):7203, 2024.

[14] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. *Advances in Neural Information Processing Systems*, 30:4768–4777, 2017.

[15] National Human Genome Research Institute. Single nucleotide polymorphism, 2024. Accessed: 2025-05-21.

[16] Bharati Panigrahi, Krishna Chaitanya Rao Kathala, and M. Sujatha. A machine learning-based comparative approach to predict the crop yield using supervised learning with regression models. *Procedia Computer Science*, 218:2684–2693, 2023. International Conference on Machine Learning and Data Engineering.

[17] Anjali Pant and R. Rajput. *Linear Regression Analysis Using R for Research and Development*, page 180:195. 10 2019.

[18] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[19] Xiaoguo Jiao Peng-Ying Wang, Zhi-Song Chen and Yun-Juan Bao. Integrating deep learning models with gwas-derived genetic variants to predict pathogenic phenotypes in group a streptococcus. *Frontiers in Microbiology*, 16:11994263, 2025.

[20] Liudmila Prokhorenkova, Gleb Gusev, Aleksandr Vorobev, Anna Veronika Dorogush, and Andrey Gulin. Catboost: Unbiased boosting with categorical features. In *Advances in Neural Information Processing Systems*, volume 31, pages 6639–6649, 2018.

[21] Luke A. Yates, Zach Aandahl, Shane A. Richards, and Barry W. Brook. Cross validation for model selection: A review with examples from ecology. *Ecological Monographs*, 93(1):e1557, 2023.

[22] Jun Zhou, Yixiao Li, and Hongbo Yang. Application of machine learning algorithms in genome-wide association studies for complex trait prediction. *Frontiers in Genetics*, 13:1041895, 2022.

UMEÅ UNIVERSITY