Interpretable Analysis of Wine Quality Using Neural Additive Models

Teo Benarous
McGill University

Abstract

Assessing wine quality often relies on subjective expert evaluations. In this work, we present an interpretable machine learning approach using Laplace-Approximated Neural Additive Models (LA-NAMs) to predict wine quality based on physicochemical properties. LA-NAMs combine the transparency of neural additive models with Bayesian uncertainty estimation, enabling credible intervals, implicit feature selection, and the discovery of feature interactions. Applied to the UCI Wine Quality dataset, our model identifies alcohol, sulphates, and volatile acidity as the most influential factors, with nuanced effects and quantified uncertainty. Additionally, interaction analysis reveals meaningful combinations of features, such as pH and fixed acidity, that contribute to wine quality. LA-NAMs achieve predictive performance comparable to state-of-the-art methods while providing robust and interpretable explanations.

1 Introduction

The wine industry often relies on expert evaluations to assess wine quality, which can be subjective. Machine learning models offer the potential to predict wine quality based on physicochemical properties, but their "black box" nature often limits interpretability. Neural additive models (NAMs), as introduced by Agarwal et al. [1], offer a promising solution by combining the expressiveness of neural networks with the interpretability of additive models. This approach allows for the decomposition of predictions into interpretable components, making it easier to understand the influence of individual features on wine quality ratings.

While NAMs address the need of transparency, they are constrained by significant limitations, including a lack of uncertainty quantification, feature selection and interaction modeling. To address these challenges, this project incorporates Laplace-Approximated Neural Additive Models (LA-NAMs), introduced by Bouchiat et al. [2], into the analysis of the Wine Quality dataset [3] from the UCI Machine Learning Repository. LA-NAMs, which extend NAMs by introducing Bayesian principles, allows for uncertainty estimates for the contributions of individual features, implicit feature selection through marginal likelihood estimation, and the identification of meaningful feature interactions. These enhancements allows for a more comprehensive understanding of the physicochemical factors that drive wine quality.

2 Related Work

Model-agnostic methods. Model-agnostic interpretability methods, such as partial dependence plots [4], SHAP [5], and LIME [6], offer some interpretability regardless of the underlying model. However, as Rudin [7] argued, their applicability to deep neural networks is limited and interpretable models should directly integrate transparency into their architecture rather than relying on post hoc explanations.

Neural additive models. Neural additive models (NAMs), introduced by Agarwal et al. [1], address the need for intrinsic interpretability by explicitly designing the architecture to model each feature separately. However, this additive and interpretable decomposition comes with trade-offs, including a lack of robust feature selection methods, and the inability to account for interactions among features

efficiently. Subsequent works have sought to address these issues (see Appendix A). Building upon these advancements, Bouchiat et al. [2] introduced Laplace-Approximated Neural Additive Models (LA-NAMs), which extend NAMs by integrating Bayesian principles. This approach enhances the interpretability and robustness of NAMs through three key innovations. First, LA-NAMs provide credible intervals for each additive sub-network, enabling uncertainty quantification. Second, using marginal likelihood estimation allows the model to perform implicit feature selection. Third, LA-NAMs rank feature pairs by using the mutual information between feature networks, allowing for efficient interaction modeling. These innovations make LA-NAMs well-suited for an interpretable analysis of wine quality.

Wine quality prediction. Wine quality prediction is not a central area of research, but the wine quality dataset is one of the most popular among the UCI Machine Learning Repository. As such, it has been widely used to empirically measure the performance of machine learning algorithms. Predicting wine quality can be seen either as a regression or a classification task. However, the few articles that address wine quality prediction as a central focus treat it as a classification task. For example, Gupta [8] conducted a comparative study of linear regression, support vector machines, and multilayer perceptrons, by categorizing the scores into “good," "average," and "bad" categories. Feature importance was assessed through statistical significance tests on the linear regression coefficients. However, this method assumes that the relationship between the target and the features is linear, potentially leading to misleading interpretations. Similarly, Jain et al. [9] approached the task as a binary classification task ("good" vs. "bad") using a decision tree, a random forest [10] and gradient-boosted decision trees [11]. Feature importance was assessed in two folds. For the decision tree and the random forest, each feature was assigned a score based on how frequently it was used to divide the data and the impurity reduction it provided. However, this method does not work well with correlated features, as importance may be distributed across multiple features, obscuring the significance of individual features. For gradient boosted decision trees, recursive feature elimination was used. However, removing features sequentially can result in suboptimal feature subsets if an early elimination drops an important feature that appears less significant initially.

3 Background

Assume that we have a dataset of the form: $\mathcal{D} = \big\{(\mathbf{x}_i, y_i) \in \mathbb{R}^d \times \mathbb{R} \,\big|\, 1 \leq i \leq n\big\}$. A neural additive model is a neural network f consisting of sub-networks $f_1, \ldots, f_d$, with parameters $\boldsymbol{\theta} = \{\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_d\}$, such that each network is applied to a single dimension, i.e.,

$$f(\mathbf{x} \,|\, \boldsymbol{\theta}) = \sum_{j=1}^{d} f_j(x_j \,|\, \boldsymbol{\theta}_j) \tag{1}$$

For simplicity, we assume that all feature networks have the same architecture. The sum is mapped to an output using a likelihood function $p(\mathcal{D} \,|\, \boldsymbol{\theta})$ and a link function g such that $\mathbb{E}[y \,|\, \mathbf{x}] = g^{-1}(f(\mathbf{x}))$. We impose a zero-mean Gaussian prior over the parameters of each feature network with prior precision hyperparameters $\boldsymbol{\lambda} = \{\lambda_1, \ldots, \lambda_d\}$, i.e.,

$$p(\boldsymbol{\theta}) = p(\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_d \,|\, \boldsymbol{\lambda}) = \prod_{j=1}^{d} \mathcal{N}\big(\boldsymbol{\theta}_d \,\big|\, \mathbf{0}, \lambda_j^{-1}\mathbf{I}\big) \tag{2}$$

These terms adaptively regularize the network parameters and enable feature selection in a similar fashion to automatic relevance determination [12]. Large values push the corresponding feature networks toward zero and low values encourage non-linear fits.

We linearize the model around a parameter estimate $\boldsymbol{\theta}^*$,

$$f^{\text{lin}}(\mathbf{x} \,|\, \boldsymbol{\theta}^*) = \sum_{j=1}^{d} f_j^{\text{lin}}\big(x_j \,\big|\, \boldsymbol{\theta}_j^*\big) \tag{3}$$

$$f_j^{\text{lin}}\big(x_j \,\big|\, \boldsymbol{\theta}_j^*\big) = f_j\big(x_j \,\big|\, \boldsymbol{\theta}_j^*\big) + \mathbf{J}_{\boldsymbol{\theta}^*}^{(j)}(x_j)\big(\boldsymbol{\theta}_j - \boldsymbol{\theta}_j^*\big) \tag{4}$$

where $\mathbf{J}_{\boldsymbol{\theta}}^{(j)} : \mathbb{R} \to \mathbb{R}^P$ is the Jacobian of the j-th feature network with respect to $\boldsymbol{\theta}_j$. This reduces the model to a generalized linear model in the Jacobians whose posterior can be approximated with

a block-diagonal Laplace approximation as:

$$q(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\theta} \mid \boldsymbol{\theta}^*, \boldsymbol{\Sigma}^*) \qquad \boldsymbol{\Sigma}^* \approx \begin{bmatrix} \boldsymbol{\Sigma}_1 & & \\ & \ddots & \\ & & \boldsymbol{\Sigma}_d \end{bmatrix} \tag{5}$$

where the diagonal covariance blocks are determined using the feature network Jacobians and second derivatives of the log-likelihood. The approximation also leads to an additive structure over feature networks in the log-marginal likelihood. Refer to Bouchiat et al. [2] for more details.

During training, the feature networks are implicitly compared and selected using adaptive regularization. This selection mechanism arises from the optimization of the lower bound on the Laplace approximation to the log-marginal likelihood [13].

Thanks to the linearization, we can obtain function-space predictive uncertainties in a closed form, like for Gaussian processes. Given an unobserved sample $\mathbf{x}^*$, the predictive variance of the linearized model corresponds to the sum of the predictive variances of the linearized sub-networks, i.e.,

$$\operatorname{Var}\left(f^{\text{lin},*} \,\middle|\, \mathbf{x}^*\right) = \sum_{j=1}^{d} \operatorname{Var}\left(f_j^{\text{lin},*} \,\middle|\, x_j^*\right) = \sum_{j=1}^{d} \mathbf{J}_{\boldsymbol{\theta}^*}^{(j)}(x_j^*)^\top \boldsymbol{\Sigma}_j \, \mathbf{J}_{\boldsymbol{\theta}^*}^{(j)}(x_j^*) \tag{6}$$

This is due to the block-diagonal structure of our posterior approximation in Equation (5). Refer to the Appendix A.1 of Bouchiat et al. [2] for more details. As training progresses, the feature networks may shift to satisfy a global intercept value in their sum. They should therefore be shifted back around zero before visualization by removing the expected contribution,

$$\hat{f}_j(x_j^*) = f_j(x_j^*) - \mathop{\mathbb{E}}_{\mathbf{x} \sim p(\mathcal{D})}[f_j(x_j)] \tag{7}$$

Note that this adjustment does not affect the predictive variance $\operatorname{Var}\left(f^{\text{lin},*} \,\middle|\, \mathbf{x}^*\right)$. This variance estimate can then be used to generate credible intervals for local and global explanations of the model. This allows the model to not only communicate on which data points it is uncertain, but also which features are responsible for the predictive uncertainty.

It is a priori unclear which features exhibit underlying interactions. As the search space grows exponentially for higher-order interactions, the authors focus on the second order. The goal is to find a subset of all existing interactions pairs that, when added to the model, maximize the gain in performance. For each selected interacting pair (j, j') we can then append a joint feature network $f_{j,j'}(x_j, x'_j)$ and perform fine-tuning of the model with the appended networks part of a secondary training stage.

The method for detecting and selecting second-order interactions makes use of the mutual information between feature networks. If the mutual information between the feature network parameters $\boldsymbol{\theta}_j$ and $\boldsymbol{\theta}_{j'}$ is high, then conditioning on the values of either of these should provide information about the other and thus, their functions. This can be an indication that a joint feature network for this pair may improve the data fit. Although the mutual information between feature networks is zero in the approximation of Equation (5), this is not necessarily the case in the true posterior. For the purpose of determining mutual information of the feature networks, we therefore fit separate last-layer Laplace approximations of the model for all feature pairs. For each candidate pair of features, the mutual information is approximated using the the scalar marginal variances and covariance in the resulting $d \times d$ posterior covariance matrix. We can then select the top-k highest scoring pairs. Importantly, this can be done after training in the first stage and without the need for an additional model to assess interaction strength. Refer to the Appendix B.2.1 of Bouchiat et al. [2] for more details.

4 Methodology

The dataset contains 11 continuous features. The outliers are handled using the IQR method. Refer to the Appendix B for more information on the dataset. Below are the explanations of the model. The global explanations are the effect of each feature on wine quality at all points, whereas the local explanations are given by the mean effect.

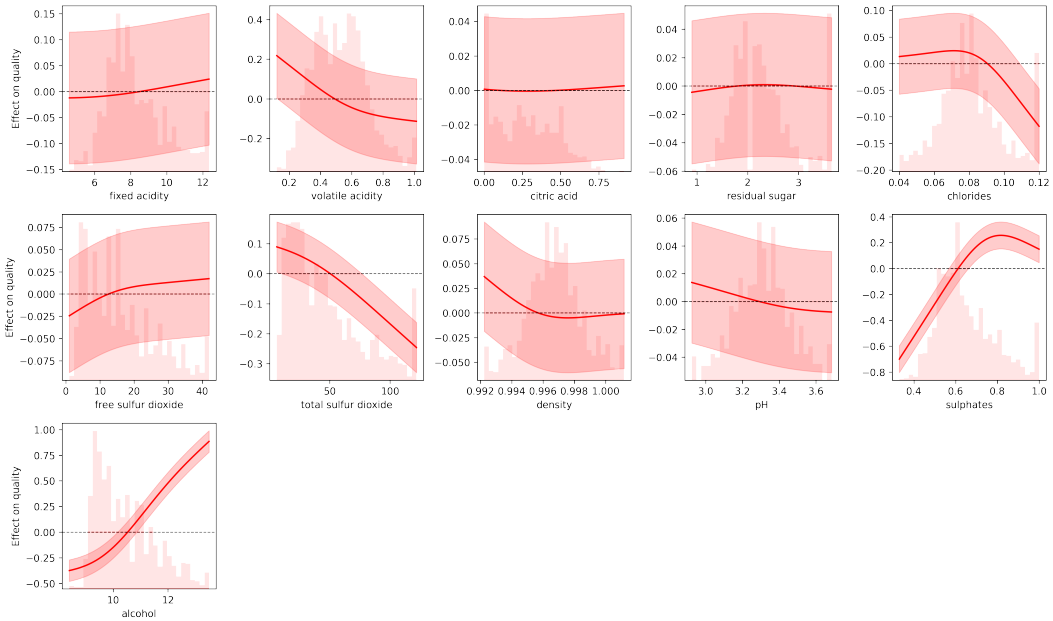


Figure 1: Effect on wine quality and associated uncertainty ($\pm$ 2 std. deviations).

We observe that the features $\{\text{fixedacidity}, \text{citricacid}, \text{residualsugar}, \text{freesulfurdioxide}, \text{density}, \text{pH}\}$ do not significantly affect wine quality. However, the credible intervals are narrower for some than others, meaning that we are more confident that some features (e.g., citric acid) do not significantly affect the quality (individually), than others (e.g., fixed acidity).

Considering the volatile acidity, for values less than 0.5 g/dm^3, the effect is positive, but decreases and eventually plateaus at a slightly negative contribution for values beyond that threshold. For chlorides levels, the contribution is around zero for values less than 0.1 g/dm^3, and then steadily decreases. For total sulfur dioxide, the contribution is also initially positive, but continuously decreases down to significant low contributions. The global explanations for sulphates exhibit an interesting pattern. The contribution is negative for values less than 0.6 g/dm^3 and steadily increases as sulphates levels increases. However, the contribution appears to plateau and slightly decrease for values beyond the 0.8 g/dm^3, indicating a diminishing return beyond that point. Alcohol is the most influential feature and initially exhibit a similar pattern as sulphates levels: for values less than 10.5% the contribution is negative and continuously increases as alcohol levels increases. In contrast to sulphates levels, the increase is more pronounced, and we do not observe a diminishing return.

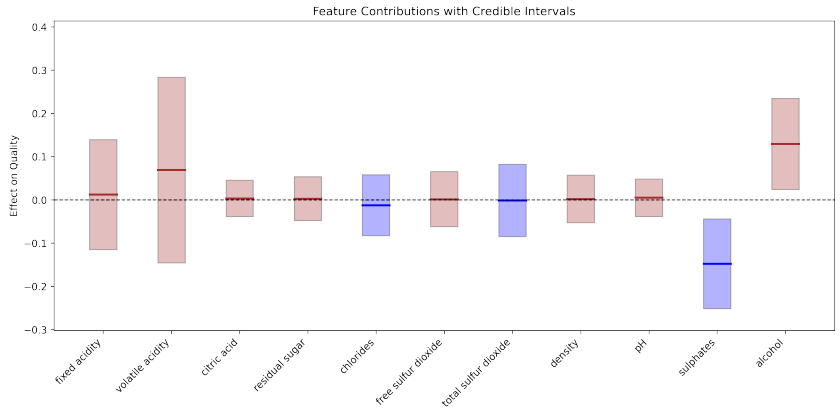
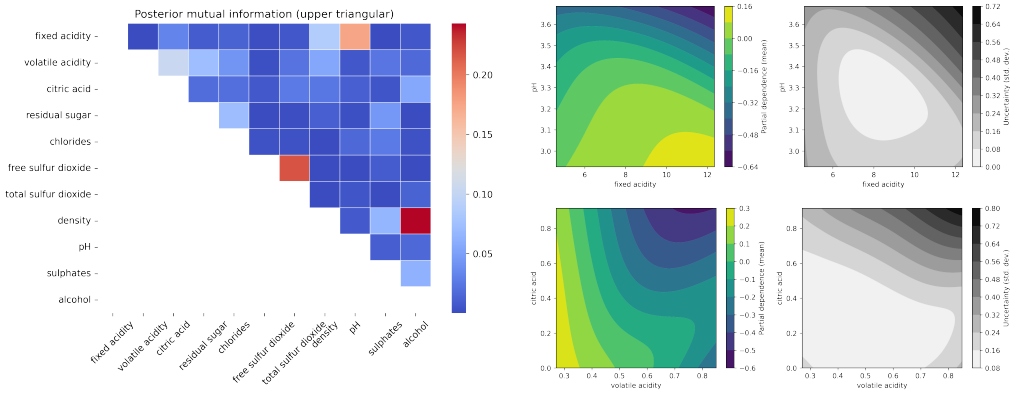


Figure 2: Local explanations (mean effect).

On average, all features except volatile acidity, sulphates and alcohol exhibit negligible effect on wine quality, and only sulphates and alcohol's credible intervals do not overlap with zero. This is particularly interesting because globally, we observe nuanced patterns for multiple features, even those with relatively narrow credible intervals locally.

The model is fine-tuned on the top five feature interaction uncovered, as shown below.



(a) (b)

Figure 3: Features interactions uncovered. (a) Last-layer posterior mutual information matrix, used to select the most informative feature-interaction pairs. (b) Two selected example feature interactions and their associated uncertainty.

The interaction pair deemed potentially most beneficial was not found to be particularly insightful. For each density level, the contribution appears similar, which coincides with our previous univariate observations. Thus, we only deduce that higher levels of alcohol tends to higher levels to wine quality. Similarly, the for pairs (low total sulfur dioxide, free sulfur dioxide) and (low volatile acidity, citric acid), the contributions only reinforced our individual observations. The contribution of (fixed acidity, density) appears negligible. Interestingly, low pH (less than 3.1) and fixed acidity levels greater than 8.5 g/dm^3 contributes to higher wine quality confidently, although both features were deemed insignificant previously. Refer to the Appendix B for figures illustrating each interaction.

Model	**RMSE**
LA-NAM	0.61
Linear	0.63
GAM	0.61
EBM	0.60
LightGBM	**0.59**

Table 1: Model performance metrics

All models achieve similar performance. Given the range of the target (commonly defined to be from 0-10 but with values in the dataset ranging from 3-8), its standard deviation (0.81), and the inherent subjectivity in taste preferences, we can assert that all models perform relatively well. The authors of the original paper conducted experiments on multiple datasets, and concluded that LA-NAMs perform competitively compared to fully interacting methods (e.g., LightGBM) and achieve similar results to gold-standard interpretable methods (e.g., EBM).

5 Conclusion

In this study, we used a LA-NAM for elucidating the factors that contribute to red wine quality. We observed that when considering the individual effect of each feature, most have a negligible effect. Whereas for some, the pattern is complex. For example, sulphates levels indicate a diminishing return beyond 0.8 g/dm^3, with a sweet spot (positive effect) for values between 0.6 g/dm^3 and 0.8

g/dm^3. Alcohol is observed to be the most influential feature, positively associated throughout its range of values with wine quality, with values greater than 10.5% no longer being associated with a negative effect.

These global explanations, gives us a more nuanced interpretation of the effect of each feature than local explanations. On average, only three features — volatile acidity (positive), sulphates (negative), and alcohol (positive) — are identified as having a significant effect, with only two of them (sulphates and alcohol) having credible intervals not overlapping with zero.

The feature interactions uncovered highlights the difficulties in interpreting what contributes to wine quality. For example, although the effect of one interaction was negligible and three reinforced our univariate observations, two features which when considered individually were deemed insignificant (pH and fixed acidity), when combined, can contribute positively to wine quality. Considering this surprising interaction and the intuition that we may expect a diminishing return on the wine quality beyond a certain threshold for alcohol levels, this motivates further research in Bayesian additive models that incorporates higher-order interactions. Another area of future work would be to expand the methodology of the original paper to use separate priors and precision terms for each layer of each feature network, instead of assuming a common architecture.

A Detailed Related Work

Neural additive models. Neural additive models (NAMs), introduced by Agarwal et al. [1], address the need for intrinsic interpretability by explicitly designing the architecture to model each feature separately. However, this additive and interpretable decomposition comes with trade-offs, including a lack of robust feature selection methods, and the inability to account for interactions among features efficiently. Subsequent works have sought to address these issues. For example, Yang et al. [14] introduced GAMI-Net, which incorporates second-order feature interactions into NAM-like architectures. Similarly, Radenovic et al. [15] proposed the Neural Basis Model (NBM), which enables scalable modeling of all possible feature interactions. Other extensions have been suggested, including feature selection through sparse regularization of the feature networks [16] and generation of confidence intervals using a spline basis expansion [17]. Kim et al. [18] addressed the inability of NAMs to model multimodal distributions, and Thielmann et al. [19] extended NAMs to capture full response distributions, including location, scale, and shape parameters.

Building upon these advancements, Bouchiat et al. [2] introduced Laplace-Approximated Neural Additive Models (LA-NAMs), which extend NAMs by integrating Bayesian principles. This approach enhances the interpretability and robustness of NAMs through three key innovations. First, LA-NAMs provide credible intervals for each additive sub-network, enabling uncertainty quantification. Second, using marginal likelihood estimation allows the model to perform implicit feature selection. Third, LA-NAMs rank feature pairs by using the mutual information between feature networks, allowing for efficient interaction modeling. These innovations make LA-NAMs well-suited for an interpretable analysis of wine quality.

Sparse and Bayesian models. Sparse additive models [20] aimed at enhancing the interpretability of machine learning models by introducing sparsity constraints. Zhao and Liu [21] extended this concept with sparse additive machines, which can be viewed as an additive functional version of support vector machines [22]. Although these models are interpretable, they do not offer uncertainty quantification and may underperform when the number of features is not significantly greater than the number of observations. Bayesian neural networks address these limitations by combining the expressiveness of neural networks with Bayesian inference [23, 24]. Various methods for approximate Bayesian inference have been explored, including variational inference [25] and ensemble methods [26]. The Laplace approximation stands out as a computationally efficient method for estimating both posterior distributions and marginal likelihoods [27]. LA-NAMs leverage the Laplace approximation to enhance NAMs with Bayesian principles, offering a novel approach for robust and interpretable machine learning.

Wine quality prediction. Wine quality prediction is not a central area of research, but the wine quality dataset is one of the most popular among the UCI Machine Learning Repository. As such, it has been widely used to empirically measure the performance of machine learning algorithms. Traditional methods, such as linear regression and decision trees, offer some interpretability but often fail to capture the complex, non-linear relationships inherent in wine quality data [28]. More advanced approaches, including support vector machines and deep neural networks, have achieved superior predictive performance, but at the expense of interpretability.

Predicting wine quality can be seen either as a regression or a classification task. However, the few articles that address wine quality prediction as a central focus treat it as a classification task. For instance, Gupta [8] conducted a comparative study of linear regression, support vector machines, and multilayer perceptrons, by categorizing the scores into “good," "average," and "bad" categories. Feature importance was assessed through statistical significance tests on the linear regression coefficients. However, this method assumes that the relationship between the target and the features is linear, potentially leading to misleading interpretations.

Similarly, Jain et al. [9] approached the task as a binary classification task ("good" vs. "bad") using a decision tree, a random forest [10] and gradient-boosted decision trees [11]. Feature importance was assessed in two folds. For the decision tree and the random forest, each feature was assigned a score based on how frequently it was used to divide the data and the impurity reduction it provided. However, this method does not work well with correlated features, as importance may be distributed across multiple features, obscuring the significance of individual features. For gradient boosted decision trees, recursive feature elimination was used. However, removing features sequentially can result in suboptimal feature subsets if an early elimination drops an important feature that appears less significant initially.

B Additional Figures

Statistic	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
mean	8.32	0.53	0.27	2.54	0.09	15.87	46.47	1.00	3.31	0.66	10.42	5.64
std	1.74	0.18	0.19	1.41	0.05	10.46	32.90	0.00	0.15	0.17	1.07	0.81
min	4.60	0.12	0.00	0.90	0.01	1.00	6.00	0.99	2.74	0.33	8.40	3.00
25%	7.10	0.39	0.09	1.90	0.07	7.00	22.00	1.00	3.21	0.55	9.50	5.00
50%	7.90	0.52	0.26	2.20	0.08	14.00	38.00	1.00	3.31	0.62	10.20	6.00
75%	9.20	0.64	0.42	2.60	0.09	21.00	62.00	1.00	3.40	0.73	11.10	6.00
max	15.90	1.58	1.00	15.50	0.61	72.00	289.00	1.00	4.01	2.00	14.90	8.00

Table 2: Summary Statistics

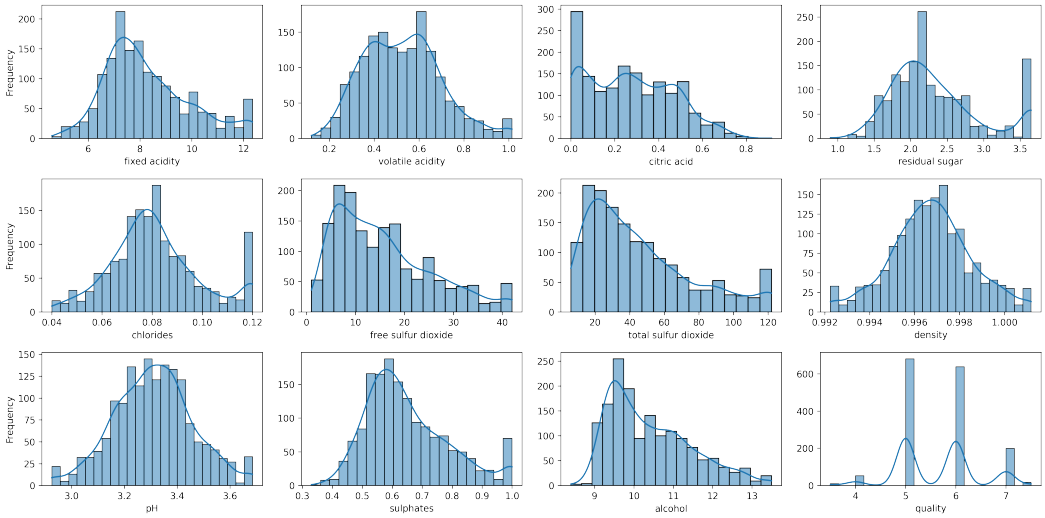


Figure 4: Distributions

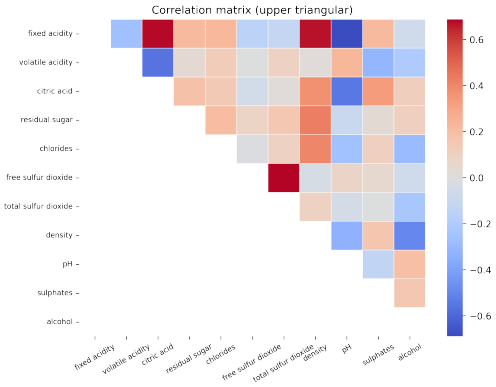


Figure 5: Correlation Matrix

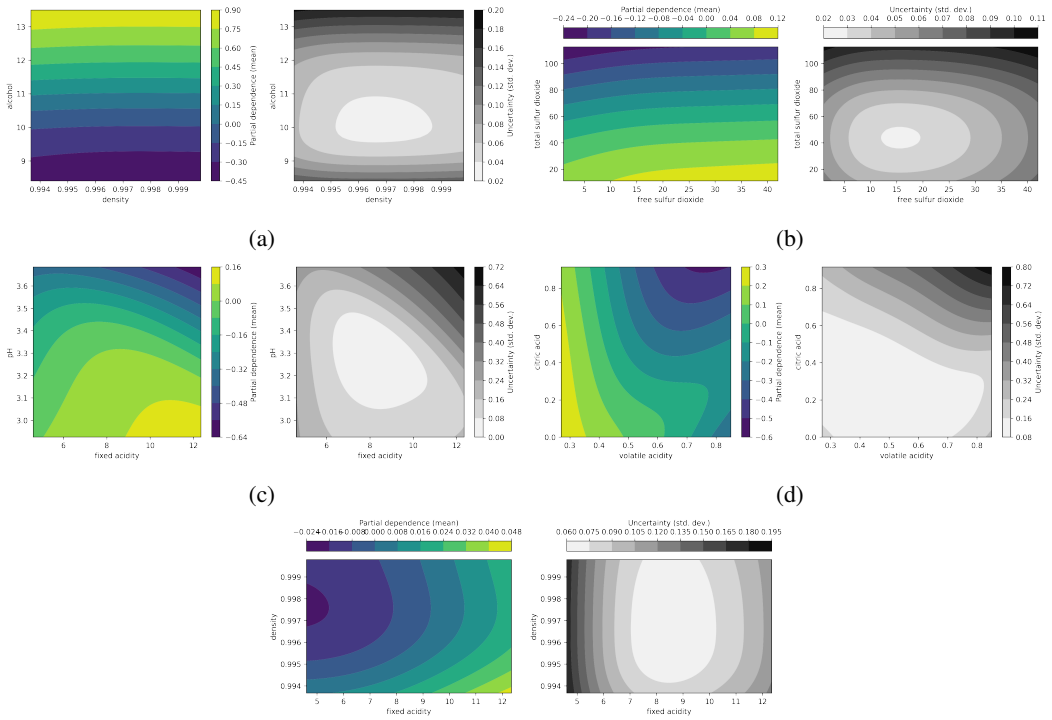


Figure 6: Feature interactions uncovered in decreasing order of approximated mutual information.

C Experimental Setup

Linear Regression. Linear regression with an ℓ_1 regularization term is implemented using the `scikit-learn` library. The regularization parameter is selected via grid search over the set $\{0.01, 0.1, 1, 10, 100\}$. Model selection is performed using 5-fold cross-validation, optimizing for the negative mean squared error.

LA-NAM. The LA-NAM is constructed using feature networks containing a single hidden layer of 64 neurons with GELU activation. Joint feature networks added for second-order interaction fine-tuning contain two hidden layers of 64 neurons. The feature network parameters and hyperparameters (prior precision and observation noise) are optimized using Adam. The learning rate is grid-searched over the set $\{0.1, 0.01, 0.001\}$ which maximizes the log-marginal likelihood. A learning rate of 0.1 is used for the hyperparameter optimization, while performing batches of 10 gradient steps on the log-marginal likelihood every 10 epochs of regular training, with a burn-in period of 25 epochs. A batch size of 32 is used.

GAM. Generalized additive models are implemented using the `pygam` library. The model is configured to use 25 splines for smoothing. The regularization parameter λ is selected through a grid search, which samples 10 candidates logarithmically spaced within the range $[10^{-3}, 10^{3}]$. The best λ is determined using generalized cross-validation scoring.

EBM. Explainable boosting machines are implemented using the `InterpretML` library. The default hyperparameters provided by the library are used. No significant improvements were observed when tuning parameters such as learning rate, maximum number of leaves, or minimum number of samples per leaf.

LightGBM. We use the open-source implementation from the `lightgbm` library. The maximum depth of each tree and the maximum number of leaves are grid-searched over $\{3, 7, 12\}$ and $\{8, 16, 31\}$, respectively. Additionally, the minimum number of samples per leaf is set to 20. Models are trained up to 500 boosting rounds, using early stopping based on validation RMSE via a 12.5% split of the training data.

References

[1] Rishabh Agarwal, Levi Melnick, Nicholas Frosst, Xuezhou Zhang, Ben Lengerich, Rich Caruana, and Geoffrey E Hinton. Neural additive models: Interpretable machine learning with neural nets. *Advances in neural information processing systems*, 34:4699–4711, 2021.

[2] Kouroche Bouchiat, Alexander Immer, Hugo Yèche, Gunnar Rätsch, and Vincent Fortuin. Laplace-approximated neural additive models: Improving interpretability with bayesian inference. *stat*, 1050:26, 2023.

[3] Paulo Cortez, A. Cerdeira, F. Almeida, T. Matos, and J. Reis. Wine quality. UCI Machine Learning Repository, 2009. DOI: `https://doi.org/10.24432/C56S3T`.

[4] Jerome H Friedman. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232, 2001.

[5] Scott Lundberg. A unified approach to interpreting model predictions. *arXiv preprint arXiv:1705.07874*, 2017.

[6] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "Why Should I Trust You?" Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144, 2016.

[7] Cynthia Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature machine intelligence*, 1(5):206–215, 2019.

[8] Yogesh Gupta. Selection of important features and predicting wine quality using machine learning techniques. *Procedia Computer Science*, 125:305–312, 2018.

[9] Khushboo Jain, Keshav Kaushik, Sachin Kumar Gupta, Shubham Mahajan, and Seifedine Kadry. Machine learning-based predictive modelling for the enhancement of wine quality. *Scientific Reports*, 13(1):17042, 2023.

[10] Tin Kam Ho. Random decision forests. In *Proceedings of 3rd international conference on document analysis and recognition*, volume 1, pages 278–282. IEEE, 1995.

[11] 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*, pages 785–794, 2016.

[12] David JC MacKay. Bayesian non-linear modeling for the prediction competition. In *Maximum Entropy and Bayesian Methods: Santa Barbara, California, USA, 1993*, pages 221–234. Springer, 1996.

[13] Alexander Immer, Tycho FA Van Der Ouderaa, Mark Van Der Wilk, Gunnar Ratsch, and Bernhard Schölkopf. Stochastic marginal likelihood gradients using neural tangent kernels. In *International Conference on Machine Learning*, pages 14333–14352. PMLR, 2023.

[14] Zebin Yang, Aijun Zhang, and Agus Sudjianto. Gami-net: An explainable neural network based on generalized additive models with structured interactions. *Pattern Recognition*, 120:108192, 2021.

[15] Filip Radenovic, Abhimanyu Dubey, and Dhruv Mahajan. Neural basis models for interpretability. *Advances in Neural Information Processing Systems*, 35:8414–8426, 2022.

[16] Shiyun Xu, Zhiqi Bu, Pratik Chaudhari, and Ian J Barnett. Sparse neural additive model: Interpretable deep learning with feature selection via group sparsity. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 343–359. Springer, 2023.

[17] Mattias Luber, Anton Thielmann, and Benjamin Säfken. Structural neural additive models: Enhanced interpretable machine learning. *arXiv preprint arXiv:2302.09275*, 2023.

[18] Young Kyung Kim, Juan Matias Di Martino, and Guillermo Sapiro. Generalizing neural additive models via statistical multimodal analysis. *Transactions on Machine Learning Research*, 2024.

[19] Anton Frederik Thielmann, René-Marcel Kruse, Thomas Kneib, and Benjamin Säfken. Neural additive models for location scale and shape: A framework for interpretable neural regression beyond the mean. In *International Conference on Artificial Intelligence and Statistics*, pages 1783–1791. PMLR, 2024.

[20] Han Liu, Larry Wasserman, John Lafferty, and Pradeep Ravikumar. Spam: Sparse additive models. *Advances in Neural Information Processing Systems*, 20, 2007.

[21] Tuo Zhao and Han Liu. Sparse additive machine. In *Artificial Intelligence and Statistics*, pages 1435–1443. PMLR, 2012.

[22] Corinna Cortes. Support-vector networks. *Machine Learning*, 1995.

[23] David JC MacKay. A practical bayesian framework for backpropagation networks. *Neural computation*, 4(3):448–472, 1992.

[24] Radford Neal. Bayesian learning via stochastic dynamics. *Advances in neural information processing systems*, 5, 1992.

[25] Alex Graves. Practical variational inference for neural networks. *Advances in neural information processing systems*, 24, 2011.

[26] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems*, 30, 2017.

[27] Alexander Immer, Matthias Bauer, Vincent Fortuin, Gunnar Rätsch, and Khan Mohammad Emtiyaz. Scalable marginal likelihood estimation for model selection in deep learning. In *International Conference on Machine Learning*, pages 4563–4573. PMLR, 2021.

[28] Paulo Cortez, António Cerdeira, Fernando Almeida, Telmo Matos, and José Reis. Modeling wine preferences by data mining from physicochemical properties. *Decision support systems*, 47(4):547–553, 2009.