Hyperparameter-Tuned Supervised Machine Learning for Alzheimer's Disease Detection: A Comparative Study on the Bias-Variance Trade-Off

Jessica Sihite
1275588
COMP90049: Introduction to Machine Learning
May 30, 2025

Introduction

To date, Alzheimer's Disease (AD) remains one of the most challenging neurological disorders, with variation in both its underlying causes and presenting symptoms. For instance, while it predominantly affects individuals aged 65 and older, not all diagnosed patients fall within this age group. Some younger patients may exhibit atypical indicators, such as abnormal Body Mass Index (BMI) or impaired cognitive test performance, without aligning with the typical age profile. This heterogeneity is well-illustrated by the Alzheimer's Prediction Dataset (Global), which comprises 50 features spanning various domains, such as demographic, physical, and cognitive (https://www.kaggle.com/datasets/ankushpanday1/alzheimers-prediction-dataset-global). At present, definitive diagnosis relies on costly and limited-access clinical procedures, such as Positron Emission Tomography (PET) scans that detect abnormal protein accumulations in the brain. Given the financial and logistical constraints of such methods, coupled with the disease's increasing prevalence, there is a pressing need for scalable, non-invasive, and cost-effective screening alternatives. This has motivated the growing interest in leveraging Machine Learning (ML) techniques to develop early-stage classification models using readily available clinical and demographic data. The present study contributes to this line of research by investigating the predictive performance of multiple ML classifiers on this multidimensional dataset, with the aim of supporting pre-clinical screening while preserving the role of PET scans as the confirmatory diagnostic standard.

Literature Review

Recent studies revealed the potential of supervised ML for early AD detection with models such as Logistic Regression and Support Vector Machine (SVM) often favoured over others due to its high accuracy scores. However, many overlook the bias-variance trade-off and its clinical consequences, particularly the balance between falsely identified negative and positive cases. To address this gap, as Cabanillas-Carbonell and Zapata-Paulini (2025) conducted a comparative framework using additional evaluation metrics, such as Precision, Recall, and F1-Score, when comparing ML models of varying variance profiles. This has inspired the present study to adopt the comparative framework using a dataset of similar format with higher dimensionality and records (Panday, 2025). The additional metrics measure how each model's variance-related tendencies impact clinical risk, thereby facilitating more informed model selection and hyperparameter tuning strategy.

Methods

Dataset

The dataset is a clean, tabular collection comprising 74,283 records of individuals, each with a dichotomous "Alzheimer's Diagnosis" class label ("Yes" or "No"). Dataset demonstrated a biased distribution to reflect real-world disparities across countries. All instances were complete, with no missing values across the 24 original features. The total number of features was expanded to 50 after a pipeline of feature pre-processing was implemented.

Feature Pre-Processing

Feature Renaming and Grouping. Features were first classified into two groups: numerical and categorical. Four features were grouped as numerical features because they store numerical values, while the remaining 20 features were assigned to the categorical group.

Manual Construction of Custom Feature ("Cognitive Test Deviation from Healthy Mean"). To reflect cognitive underperformance among individuals with AD relative to healthy individuals, a custom numerical feature was engineered from a raw feature "Cognitive Test", which represents individual score on a cognitive test. Mean "Cognitive Test" scores among instances with class label "No" was computed. Subtraction of each "Cognitive Test" by the mean "Cognitive Test" of healthy individuals was subsequently implemented.

Polynomial Feature Construction. All numerical features, including the engineered feature, underwent second-degree polynomial feature expansion, combining any two of numerical features to capture potentially nonlinear interaction effects. The approach involved squaring each numerical value and obtaining cross-product terms between numerical feature pairs.

Numerical Value Standardization. All numerical features, including the expanded features, were then standardized to ensure all features contributed fairly to the model's decision boundary. This step is particularly important for algorithms that are sensitive to feature scale, such as Logistic Regression and *k*-NN.

One-Hot Encoding. For each categorical feature, the most frequent categorical value weas one-hot encoded, transforming it into a separate binary feature.

Feature Selection

To reduce dimensionality while retaining predictive power, all features were ungrouped and univariate filtering using the Information Gain criterion was applied. This filter-based method scored each feature based on its individual association with the class label for each instance. In this study, only 50 features with the highest Information Gain value were selected. As per standard practice, feature selection was nested within the training folds of the cross-validation pipeline.

Cross-Validation Technique

Cross-validation was performed independently for each classification algorithm to ensure a fair and isolated evaluation of model performance. For every model, the dataset was split into multiple folds, with training and testing strictly separated throughout the process. This was subsequently followed by nested cross-validation for each fold.

Outer Fold

A 6-fold stratified cross-validation partitioned the dataset into six outer subsets (folds), such that the class distribution in each fold remains representative of the class distribution in the original dataset. Each fold served as the outer test subset once, while the remaining five were used for training. Therefore, classification for each model was done for six iterations.

Nested Cross-Validation: Inner Fold

Within each of the five training folds, an inner 3-fold cross-validation was performed to select 50 best features and evaluate multiple hyperparameter combinations.

Feature Selection. Feature selection was conducted independently using the inner training folds of each outer training set, selecting 50 features with the highest Information Gain. As a result, selected features varied across folds. This ensured feature selection process was strictly based on training data and did not influence the outer test fold, maintaining the integrity of cross-validation process.

Hyperparameter Tuning. Hyperparameter tuning was conducted within the inner loop of nested cross-validation to avoid overfitting and data leakage. For each classifier, a grid search strategy was employed to systematically evaluate all pre-defined value combinations of the selected hyperparameters on the inner training folds. This procedure ensured the hyperparameters were optimized independently of the outer test fold, preserving the integrity of model evaluation. In this study, the evaluation metric scores for all hyperparameter combinations were computed per fold in each classifier.

Classification Algorithm

Three supervised ML models, varying in bias and variance, were explored in terms of their interpretability, practicality, and distinct learning mechanisms that allow insightful comparisons.

Logistic Regression

Logistic Regression estimates the probability of a given instance's class membership by running the logistic function, which maps input features to a value between 0 and 1. By applying a decision threshold to this output, the model can assign a class label in binary classification tasks. The classifier is known for its low variance, as it consistently produces similar results across different training folds; however, it may oversimplify the data due to its high bias, which stems from its assumption of linearity. Given the large dataset size and the considerably higher dimensionality due to the feature pre-processing, *saga* optimization algorithm was selected when running Logistic Regression. Model performance was further improved through hyperparameter tuning, targeting one key hyperparameter.

Inverse of Regularization Strength (*C*). To control the bias-variance trade-off in highly dimensional data, a smaller *C* value indicates stronger regularization by penalizing large coefficients, thereby reducing overfitting. Conversely, a larger C may cause the model to closely fit the training data, causing overfitting.

Decision Tree

The Decision Tree classifier places each instance within a feature space, then recursively partitions the space by selecting optimal decision thresholds that best separate the classes. This approach demonstrates low bias. Each internal node represents a decision rule based on a feature value, dividing the fold into increasingly homogenous subsets with respect to the target class, which is ultimately assigned at each leaf node. Because of the ability to closely fit the training data, Decision Tree is sensitive to noise and small fluctuations in the data, demonstrating high variance. To mitigate overfitting and improve generalizability, two hyperparameters were tuned using nested cross-validation.

Maximum Depth. The hyperparameter controls maximum number of splits from the root to a leaf node. Shallower trees tend to underfit due to their limited representational capacity, resulting in high bias; while deeper trees capture intricate patterns at the risk of overfitting to noise in the training data, resulting in high variance. Tuning this hyperparameter allowed the model to achieve an optimal complexity with a balanced the bias-variance trade-off.

Minimum Sample Split. The hyperparameter defines the minimum number of training instances required to split an internal node. Larger values introduce regularization by limiting the model's growth, thereby reducing variance; whereas lower values may create overly specific splits, causing the model to overfit and generalize poorly to unseen data.

k-Nearest Neighbours (k-NN) Classifier

The *k*-NN classifier is an instance-based learner, assigning a class label to an instance based on the majority label of its *k* closest training samples in the feature space. In this study, a default Euclidean distance metric was used to calculate the proximity between instances. To account for the complex class boundaries and uneven data distribution, common characteristics in medical diagnosis tasks, distance-weighted voting was applied. This approach reduces misclassifications near class boundaries by assigning greater classification influence to instances of closer proximity. The weighting was specified and not tuned because numerical features had been standardised, revealing negligible performance difference between uniform- and distance- weighting in preliminary evaluations. As a highly variant classifier, *k*-NN makes no assumption about the underlying data distributions, making it flexible for datasets with complex and non-linear patterns. However, such flexibility renders it highly sensitive to the choice of hyperparameters, one of which was observed in terms of the bias-variance trade-off.

Number of Neighbours (*k*). Because *k*-NN is non-parametric, it makes no assumption about the data distribution, making it highly flexible but also sensitive to the number of neighbouring instances selected for classification decision. This hyperparameter determines how many neighbouring instances contribute to the classification decision of an instance. A smaller *k* value typically leads to high variance and overfitting, as the model becomes overly sensitive to noise in the training data. Contrastingly, larger *k* reduces variance but increases bias by overly smoothing decision boundaries in a broader local context, which may ignore intricate patterns in the data.

Results and Discussion

Complete tuning results across four evaluation metrics for each hyperparameter combination are presented in Table 1 (see Appendix). These results reflect both the expected theoretical trade-offs and the empirical consequences of applying each model on the dataset.

Logistic Regression

As shown in Table 1, Logistic Regression yielded competitive performance across all evaluation metrics, with lower Standard Deviation (*SD*) indicating relatively stable results across folds. The results were consistent with the model's low-variance nature, making it reliable for larger datasets. Furthermore, its ability to produce interpretable probability scores is particularly valuable in high-stakes contexts such as medical diagnosis. In the case of AD detection, a patient classified as "No" may still exhibit a moderate probability belonging to the "Yes" class, thereby prompting early intervention or further clinical assessment.

Despite the advantages, Logistic Regression is characterized by high bias, primarily due to its strong linearity assumption of the relationship between input features and the log-odd values of the class. This simplification may become problematic when features in dataset exhibit complex, non-linear interactions, as in the case with the Alzheimer's Prediction Dataset (Global). The limitation is further compounded by its sensitivity to feature scale, potentially skewing the coefficient estimates if not properly managed. Consequently, the model may oversimplify the decision boundary and fail to capture subtle, decision cues. This inherent bias was evident in the model's comparatively low Precision, the lowest among reported evaluation scores for Logistic Regression. This suggests the bias may have encouraged the model to assign relatively higher log-odd values for "Yes" labels in each instance. It may also explain the high Accuracy scores, demonstrating a high number of correctly identified "Yes" labels. However, the model may have classified many false positives, predicting AD when it was not present. This tendency also contributed to higher average Recall scores across all hyperparameter combinations, outperforming those of k-NN and a subset of Decision Tree configurations. This indicates that Logistic Regression successfully identified a substantial proportion of actual "Yes" instances, relatively higher than the number of incorrectly identified "Yes" labels. In medical diagnosis, where false negatives cost more than the false positives, the recall-oriented behaviour may be preferable. One possible explanation lies in the robust feature pre-processing pipeline. Polynomial feature expansion introduced second-order interactions, categorical features were appropriately encoded, and standardization ensured a uniform feature scale. Collectively, these processes likely enabled Logistic Regression to approximate non-linear patterns within the transformed space, compensating for its inherent linear constraints.

With respect to regularization, lower *C* values yielded consistently higher evaluation scores, thereby reducing overfitting and reinforcing high bias. An exception emerged in the F1-Score, which peaked at a *C* value of 1. This suggests a more favourable balance between Precision and Recall under reduced regularization pressure, reinforcing the F1-score's sensitivity to the trade-offs between both metrics.

Decision Tree

The Decision Tree classifier produced the most variable results, reflecting its high variance and susceptibility to overfitting, particularly when trained on a high-dimensional dataset of over 70,000 records. This variability stems from the tree's greedy and recursive partitioning strategy which captures local patterns without global regularization. Therefore, Decision Tree was selected as the representative high-variance model in this study, offering a contrast to the stable yet biased nature of Logistic Regression and the distance-based flexibility of *k*-NN. Due to its high variance, Decision Tree exhibited the highest sensitivity to hyperparameter configurations among the three evaluated models. This was particularly due to inclusion of two tuned hyperparameters, unlike the single-hyperparameter tuning in the other two algorithms.

Among its own evaluation metrics, Recall was consistently the highest, indicating the model's stronger tendency to detect actual "Yes" labels rather than misclassifying them. However, its Recall still underperformed Logistic Regression, suggesting that while Decision Tree could model complex class boundaries, their generalizability remained limited compared to Logistic Regression. This was possibly due to fragmentation from recursive splits. Contrastingly, Precision was the lowest both among all evaluation metrics reported for Decision Tree and across all classifiers, highlighting that a significant proportion of instances classified as "Yes" were in fact false positives. This behaviour aligns with the model's tendency to overfit to noise, leading to aggressive partitioning and spurious splits especially when depth is not constrained. This also explains the lower scores for all evaluation metrics with higher maximum depth. In clinical contexts, this undermines the reliability of predictions, raising concerns about unnecessary stress or diagnostic costs for patients incorrectly flagged as positive. However, minimum number of samples for splits as the other parameter, yielded less consistent trends. While smaller values for splits were selected for finer-grained partitions and to reduce the risk of missing rare but clinically relevant patterns, these settings seemed to contribute to unstable performance across folds. Given the large dataset size, larger values may have introduced more helpful regularization by suppressing unnecessary branching.

The combined tuning of two hyperparameters contributed to the wide variability observed across folds. Similar to Logistic Regression, Decision Tree detected a significant number of false positives, although it was driven more by overfitting than by class leniency. However, Decision Tree also produced a higher number of false negatives compared to those detected by Logistic Regression. Although the classifier did not product the worst results in any one metric among all evaluated models, it also failed to exhibit dominance or consistency across metrics. The model's limited robustness and sensitivity to tuning undermined its reliability in high-stakes classification tasks such as AD detection. Therefore, Decision Tree requires careful balancing between underfitting and overfitting, which becomes particularly challenging in datasets with complex feature interactions.

***k*-Nearest Neighbour (*k*-NN) Classifier**

Following the high-bias generalization of Logistic Regression and the high-variance expressiveness of Decision Trees, the *k*-NN offers a fundamentally approach as a non-parametric, instance-based learning balancing flexibility with computational simplicity. The *k*-NN classifier was selected as a substitute for SVM, which was initially considered in this study for its robustness in modelling high-dimensional, non-linear decision boundaries through kernel functions. However, preliminary experimentation revealed SVM became computationally infeasible given the substantial dataset size and the overhead of nested cross-validation pipeline. In contrast, *k*-NN served as a practical alternative, capable of capturing non-linear

relationships with manageable computational cost despite a fundamentally different learning mechanism. To accommodate *k*-NN's sensitivity to feature scale, pre-processing pipeline involved standardisation of all numerical values. This prevented high-magnitude features, such as patients' cognitive test performance, from dominating smaller-scale yet diagnostically relevant features, such as patients' age.

As shown in Table 1, *k*-NN's performance was highly dependent on the number of neighbours (*k*), reflecting its inherent susceptibility to the bias-variance trade-off. A pattern was shared in most evaluation metrics, where performance improved with increasing *k* values before slightly declining at the highest observed setting ($k = 767$). Smaller *k* values tend to yield highly flexible models with low bias but high variance, but larger *k* values reduce variance at the cost of increased bias. While this initially improved generalization, excessively high *k* values caused the model to over-smooth the decision boundary, obscuring nuanced patterns necessary for detecting minority class instances. An exception to this trend was observed in Precision's continued growth with *k* values. This implies a larger *k* resulted in a more conservative model in predicting positive class ("Yes"), thereby reducing false positives at the expense of underfitting. Indeed, *k*-NN yielded the highest Precision among all classifiers. This behaviour may be beneficial in clinical settings because when it predicted an individual as having AD, it was more likely to be correct. Therefore, unnecessary stress or costly clinical follow-ups for healthy individuals could be avoided. This also translated into a higher average Accuracy, making it the strongest among *k*-NN's evaluation metrics. Accuracy in *k*-NN remained lower than that of Logistic Regression, partly due to the model's conservative behaviour.

However, the approach to opt for certainty over coverage also resulted in fewer true positives relative to false negatives. This was indicated by the lowest Recall among all evaluation metrics for *k*-NN and among all Recall scores reported across classifiers. Notably, Recall scores also varied by SD across hyperparameter settings compared to other metrics. This indicates performance instability in capturing positive class depending on the *k* value. Furthermore, it also underscores the difficulty of balancing recall-oriented sensitivity and precision-oriented conservatism in instance-based learning. Ultimately, while *k*-NN offered a practical workaround for SVM's computational demands, its strengths lied in reducing false positives, while its weaknesses emerged in failing to detect actual AD cases. The trade-off suggests a higher rate of missed actual AD cases, a serious limitation in medical contexts where underdiagnosis is detrimental.

Conclusion

This study highlighted the trade-offs between bias-variance and clinical priorities in AD classification. With relatively low variance but high bias, Logistic Regression achieved the highest scores across all evaluation metrics among all models, except Precision. This indicates the model struggled in correctly classifying healthy individuals, which may lead to unnecessary stress and additional cost for confirmatory clinical screening. Contrastingly, Decision Tree offered more variable results given its high variance and demonstrated overfitting, yielding high number of both false positive and false negative classes. Finally, *k*-NN yielded the highest Precision but the lowest Recall among all classifiers, demonstrating conservative behaviour. While this approach reduced the risk of false positive cases, it raised a serious clinical concern over underdiagnosis. These outcomes underline that model selection should weigh not only statistical performance but also clinical consequences of misclassification. While Logistic Regression emerged the most suitable classifier in this study, future work should investigate improved hyperparameter tuning or hybrid models to balance sensitivity and specificity in high-stakes diagnostic tasks.

References

Cabanillas-Carbonell, M., & Zapata-Paulini, J. (2025). Evaluation of machine learning models for the prediction of Alzheimer's: In search of the best performance. *Brain, Behavior, & Immunity – Health*, *44*, 100957. https://doi.org/10.1016/j.bbih.2025.100957

Panday, A. (2025). *Alzheimer's Prediction Dataset (Global)* [Data set]. Kaggle. https://doi.org/10.34740/KAGGLE/DSV/10618775

Appendix

Table 1

Mean and Standard Deviation of Evaluation Metrics Across Hyperparameter Combinations for Each Classifier

Classifier {Hyperparameter}	Hyperparameter Value	Accuracy		F1-Score		Precision		Recall	
		M	*SD*	*M*	*SD*	*M*	*SD*	*M*	*SD*
Logistic Regression {*C*}	0.1	.713	.001	.678	.001	.631	.001	.732	.001
	1	.713	.001	.681	.001	.630	.001	.741	.001
	10	.712	.001	.680	.001	.629	.001	.741	.002
Decision Trees {Maximum Depth; Minimum Sample to Split}	{5; 2}	.721	.003	.695	.003	.634	.007	.771	.017
	{5; 5}	.712	.002	.696	.002	.632	.004	.776	.013
	{5; 10}	.721	.003	.695	.003	.634	.006	.771	.015
	{10; 2}	.711	.002	.680	.002	.626	.003	.744	.008
	{10; 5}	.712	.001	.682	.001	.625	.003	.751	.006
	{10; 10}	.709	.002	.680	.003	.624	.004	.749	.011
	{20; 2}	.667	.004	.616	.007	.589	.004	.647	.012
	{20; 5}	.665	.003	.619	.005	.584	.004	.658	.104
	{20; 20}	.667	.005	.625	.006	.585	.005	.671	.011
k-Nearest Neighbour {*k*}	5	.671	.002	.577	.002	.616	.004	.541	.003
	55	.708	.001	.626	.004	.665	.001	.590	.006
	99	.711	.001	.633	.002	.667	.002	.601	.004
	389	.713	.001	.635	.001	.669	.002	.604	.003
	767	.712	.001	.632	.002	.671	.001	.597	.003

Note. Metrics are computed across six outer folds of nested cross-validation for each hyperparameter combination.