Supervised Learning Model Comparison

Nikhil Solanki
nikhil.u.solanki@gmail.com

February 2026

1 Introduction

This report analyzes several supervised learning algorithms on different datasets. This report investigates the performance and behavior of these algorithms on these datasets, with varying characteristics. The goal of this is to understand how model complexity, dataset properties, and hyperparameter tuning combine to demonstrate effectiveness of these models in the real-world.

The algorithms implemented were Decision Trees, Support Vector Machines (Linear + RBF), k-Nearest Neighbors, and Neural Networks (scikit-learn + PyTorch). These algorithms represent different model families with varying complexities, biases, and computational characteristics. Each model was evaluated on both datasets using model complexity curves, learning curves, timing analysis, and confusion matrices. Performance was primarily measured using F1 score and additional metrics such as accuracy. By understanding these metrics and the inherent tradeoffs of model performances, we can understand the bias/variance tradeoff of model selection in the real-world.

2 Exploratory Data Analysis and Hypotheses

Each algorithm was implemented on the Adult Income (Census Income) dataset and Wine Quality dataset (red + white). The Adult dataset represents a larger, real-world binary classification problem with a class imbalance (3:1); income level for people $\leq 50k$ or $> 50k$. It consists of over 45,000 entries and mix of categorical and numerical features. The target is *income*, a binary and imbalanced variable. This motivates the use of preprocessing techniques like one-hot encoding and feature scaling (standardization) to ensure compatibility for all models.

The Wine dataset represents a smaller multiclass classification problem, with the the target of classifying wines based on their *quality* ratings. It consists of roughly 6500 entries, of solely numerical data describing aspects of the wine. The class distribution is favored towards mid-quality ratings, which models must balance sensitivity on.

Based on this analysis, it is hypothesized that: (Adult Dataset)–**Hypothesis 1**: As decision tree depth increases, the model will peak validation performance at moderate depths, and decline at higher depths due to overfitting, since deeper trees capture noise in high-cardinality categorical features. **Hypothesis 2**: As the number of neurons per layer of the Multilayer Perceptron (NN) increases, the validation performance will increase under Stochastic Gradient Descent (SGD) to an extent, as wider networks capture more feature interactions, but performance will eventually plateau or decline due to overfitting and optimization limitations. (Wine Dataset)– **Hypothesis 1**: As the number of neighbors (k) increases in k-Nearest Neighbors, the validation performance will improve initially due to smoother decision boundaries, larger k values will eventually bias towards the dominant middle classes, reducing overall performance. **Hypothesis 2**: Due to the dataset exhibiting nonlinear and overlapping class structure, a Linear Support Vector Machine (SVM) is expected underfit across regularization settings, whereas a Radial Basis Function SVM is expected to achieve higher validation performance at intermediate values of C and γ, with low values leading to underfitting and high values leading to overfitting.

3 Experiments

3.0.1 Methods

Model performance was evaluated using F1 score, with hyperparameters selected based on stratified k-fold cross-validation again (5-fold for Adult, 3-fold for Wine (using built-in sklearn functions). Cross-validation was used throughout model tuning and evaluation to estimate the generalization and support any bias-variance tradeoffs across the various experiments ran.

For the PyTorch MLP implementation, hyperparameter tuning was conducted using a stratified train/validation split (70% training, 30% validation) as opposed to cross-validation due to the added complexity of implementing cross validation since PyTorch does not have its own built-in cross-validation functions. A fixed random seed (66) was applied over all Python, NumPy, and Pytorch, to confirm reproducibility.

3.1 Adult Dataset

The Adult dataset experiments evaluate how different supervised learning algorithms perform on a high-dimensional dataset, with both categorical and numerical features. Performance was analyzed using model complexity curves, learning curves, timing analysis, and confusion matrices, to understand the bias-variance tradeoffs and classification behavior. Results are interpreted in regards to the aforementioned hypotheses.

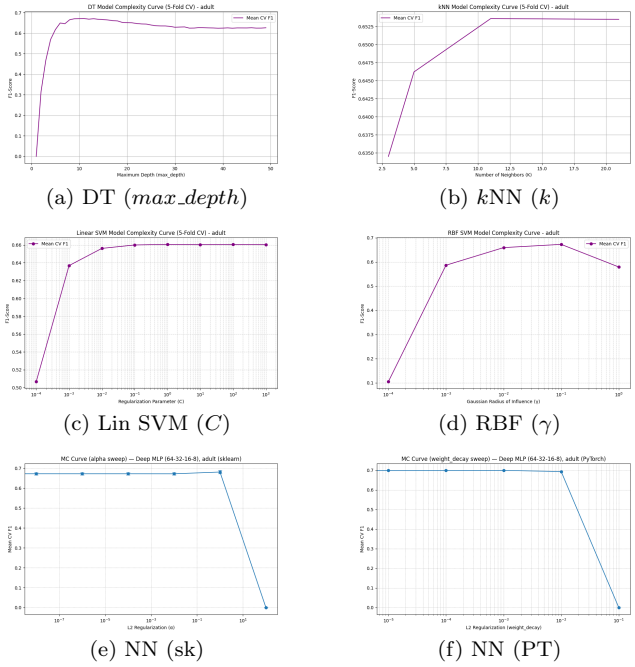
3.1.1 Model Complexity Curves

Model complexity was evaluated by varying key hyperparameters for each algorithm (tree depth for Decision Trees, number of neighbors for KNN, regularization parameters for SVM, weight decay/capacity for Neural Networks). Mean cross-validation F1 scores were plotted to analyze the bias-variance tradeoffs and identify optimal hyperparameter tuning. The Decision Tree depicts a rapid increase in F1 score as the max depth increases from shallow values, peaking around $max\_depth = 7$ before declining. This indicates that shallow, less complex trees (high bias) tend to underfit while deeper trees tend to overfit. Finding a moderate tree depth provides the best tradeoff between model flexibility and generalization.

The kNN curve depicts an expected bias-variance tradeoff with respect to the neighborhood size (adjusted by k). Small k values produce lower validation performance due to high variance and sensitivity to local noise. While performance stabilizes around $k = 11$, predictions become more robust through the neighborhood averaging. Larger k values do not improve performance further, highlighting an increasing bias as neighborhoods become too large.

For SVMs, increasing C improves performance initially by reducing regularization and fitting the decision boundary better, with validation performance eventually reaching a plateau (around $C = 10^0$). This is indicative of the model capturing the underlying linear structure of the data, supporting less returns at higher C values. The RBF kernel shows a similar pattern where performance improves initially until around a peak of $\gamma = 10^{-1}$. After this, validation performance declines due to model overfit because decision boundaries becomes overly sensitive to the training points.

For both Neural Network implementations in scikit learn and PyTorch, the $L2$ regularization hyperparameter (α - sklearn, *weight_decay* - PyTorch) strongly impacts validation F1 performance. They both demonstrate the strong sensitivity to regularization strength. Performance remains generally stable across small α or *weight_decay* values, but collapses after $\alpha = 1$ and $weight\_decay = 0.001$. This indicates excess weight constraints limit model capacity and hinder effective learning.

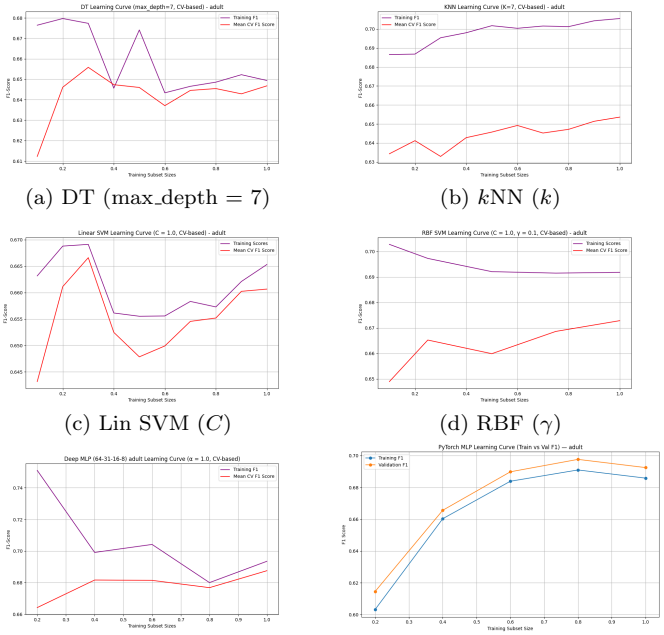


(a) DT (max_depth) (b) kNN (k)

(c) Lin SVM (C) (d) RBF (γ)

(e) NN (sk) (f) NN (PT)

Figure 1: Model complexity curves (Adult).



(a) DT (max_depth = 7) (b) kNN (k)

(c) Lin SVM (C) (d) RBF (γ)

(e) NN (sk) (f) NN (PyT)

Figure 2: Learning curves for all models the Adult dataset. **Purple** = Training F1 score, **Red** = Mean CV F1 score.

3.1.2 Learning Curves

Learning curves for all algorithms and both datasets, were generated by training each model on progressively larger subsets of the original training data ($\approx 10\% - 100\%$).

Decision Tree ($max\_depth = 7$): Training and validation F1 scores converge quickly and remain relatively close, indicating low variance at the set depth. Both curves plateau early with modest training performance, suggesting mild underfitting (bias-limited). Increasing data alone yields limited improvement unless model capacity is increased (deeper trees), which the MC curve suggests ould reintroduce overfitting.

kNN ($k = 11$): Higher training F1 score than validation F1 score, showing moderate variance likely due to sensitivity to local neighborhoods. Training performance steadily increases with more data while validation performance improves marginally, suggesting some benefit from additional data (stabilize neighborhoods) but the remaining gap suggests variance as a limiting factor.

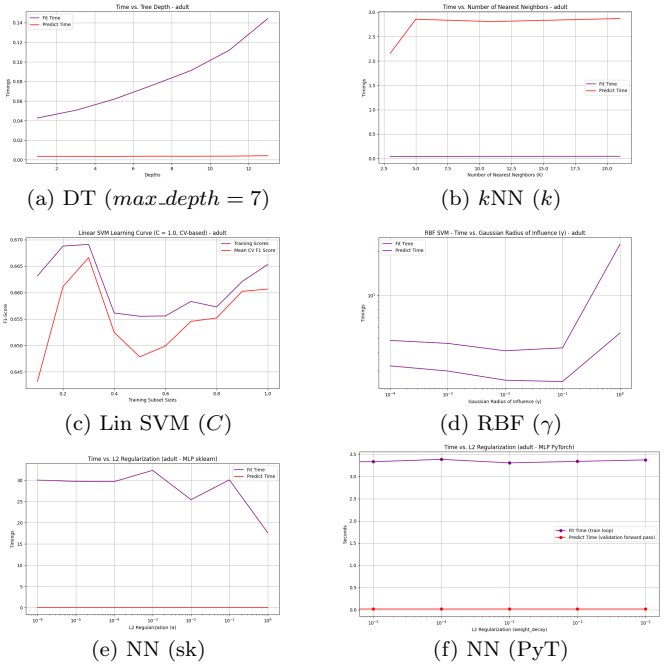
LinearSVM ($C = 1.0$): Training and validation F1 scores very close across all subset sizes, indicating low variance and better generalization. Performance stabilizes quickly suggesting the model captures most of the linear structure early in training, with the plateau suggesting bias-limited model behavior. **RBF SVM** ($\gamma = 0.1$): Training performance slightly decreasing as training sub-

set size increases, while validation performance steadily improves. This is indicative of reduced overfitting with more data. The persistent gap between the curves suggest the model maintains low variance due to the nonlinear decision boundary, leading to better generalization. Performance stabilizes quickly, suggesting the model captures most of the linear structure early in training but must be carefully controlled to prevent overfitting.

Neural Networks (sklearn Deep MLP (64-31-16-8; $\alpha = 1.0$), PyTorch Deep MLP (64-31-16-8; $\alpha = 0.001$): Both neural network learning curves show improved validation performance as the training size increases, suggesting the model benefits from additional data to generalize (and reduce overfitting). The sklearn MLP portrays higher initial training F1 performance that declines as training size increases, suggesting less "memorization" and improved generalization with more data. The PyTorch MLP shows stable convergence at larger training sizes while maintaining a small gap. The narrow gap between the two lines suggest the model benefits from more data to improve generalization, pointing to controlled model capacity and effective regularization. The difference does also indicate moderate variance, depicting the sensitivity of neural networks to training data size and regularization strength.

3.1.3 Timing Curves

To evaluate each model's computational efficiency, fit time and prediction time was measured while varying the aforementioned key hyperparameters for each model and for both datasets. These curves illustrate how model complexity influences computational cost and highlight any tradeoffs between training overhead and inference efficiency. This was conducted on a Apple MacBook Pro with the M4 Max chip.



(a) DT ($max\_depth = 7$) (b) kNN (k)

(c) Lin SVM (C) (d) RBF (γ)

(e) NN (sk) (f) NN (PyT)

Figure 3: Timing curves for all models the Adult dataset. **Purple** = fit time, **Red** = predict time.

Decision Tree fit time increases steadily as depth increases due to the growing number of splits evaluated during training. Prediction time remains relatively fast since inference on the model only takes a single tree path (root to leaf). Linear SVM also shows a gradual increase in training time as the regularization weakens (larger C), with a negligible stable prediction time. This indicates efficient inference once the model learns the linear decision boundary.

On the contrary, the kNN portrays minimal training cost but much higher prediction time, indicative of the computational burden of distance calculations at inference. The RBF SVM portrays higher training overhead due to kernel evaluations at scale (as well as complexity of fitting nonlinear decision boundaries), with prediction time increasing as γ increases (more support vectors contributing to predictions). Neural network training portrays higher fit times due to the iterative optimization across many epochs. Prediction times for both models remain very low, indicating efficient forward passes once the parameters are learned. Overall, these results point to the expected tradeoff between model flexibility and computational overhead. Simpler models provide faster inference, more complex models require greater training.

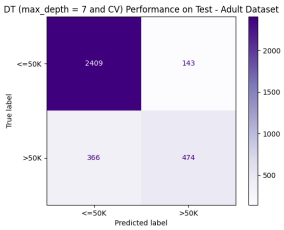
3.1.4 Confusion Matrix

For all models, confusion matrices reveal consistent patterns drive by the Adult dataset's large class imbalance. The $\leq 50K$ class is more prevalent in the dataset than the $> 50K$ class. Every model achieves strong performance in correctly identifying the majority class, with high true positive and high true positive counts. There is subsequently more difficulty in identifying the minority class which explains the higher false negative counts across models.

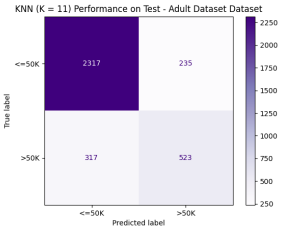
The Decision Tree demonstrates solid overall performance with a strong identification of the $\leq 50K$ class, and moderate recall for the $> 50K$ class. The balanced distribution between false negatives and false positives suggest a stable and more conservative decision boundary. Similarly, the Linear SVM shows similar behavior with favoring simpler decision boundaries, by maintaining the strong majority class performance while slightly under-predicting the minority class. kNN also shows a similar distribution of classification, but has slightly more increased false positives, highlighting its sensitivity to local neighborhood structure. This is representative of how instance-based methods (like kNN) can capture local patterns while still inflicting some variance.

The RBF SVM improves minority class detection slightly more than its linear counterpart. This reflects the benefit of its ability to model nonlinear relationships. However, there is a gap between true positives and false negatives which indicates the existing challenge of class imbalance even with an increase in model flexibility.

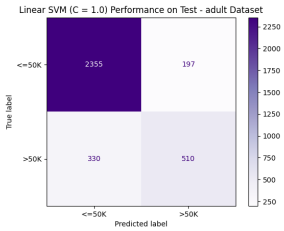
Both neural network implementations demonstrate the strongest ability to capture the minority class. The PyTorch MLP achieves high true positive count with less false



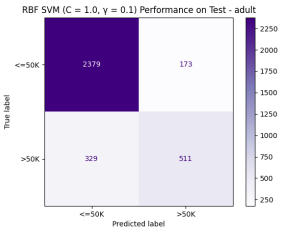
(a) DT ($max\_depth = 7$)



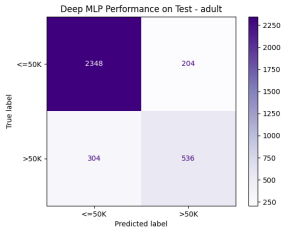
(b) kNN (k)



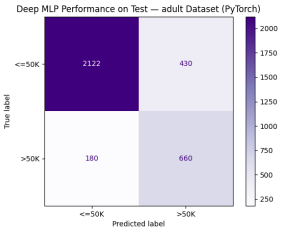
(c) Lin SVM ($C = 1.0$)



(d) RBF (γ)



(e) NN (sk - $a = 1.0$)



(f) NN (PyT - $weight\_decay = 10^{-3}$)

Figure 4: Confusion matrices for all models the Adult dataset.

negatives, highlighting its improved recall for the minority class. This suggests that a neural network's high capacity and nonlinear modeling potential enable them to capture more complex feature interactions in the data. The sklearn MLP shows similar but slightly more conservative behavior, balancing minority and majority class detections. Overall all models perform well on the majority class ($\leq 50K$), but the neural networks and SVMs outshine in minority class ($> 50K$) detection.

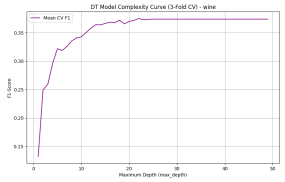
3.2 Wine Dataset

The Wine dataset experiments evaluate how different supervised learning algorithms perform on a smaller, multiclass dataset, with all numerical features. Performance was analyzed using model complexity curves, learning curves, timing analysis, and confusion matrices, to understand the bias-variance tradeoffs and classification behavior. Results are interpreted in regard to the aforementioned hypotheses.

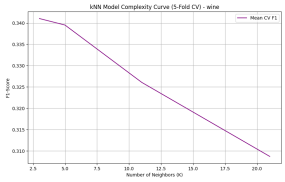
3.2.1 Model Complexity Curves

Model complexity was evaluated by varying key hyperparameters for each algorithm (tree depth for Decision Trees, number of neighbors for KNN, regularization parameters for SVM, weight decay/capacity for Neural Networks). Mean cross-validation F1 scores were plotted to analyze the bias-variance tradeoffs and identify optimal hyperparameter tuning.

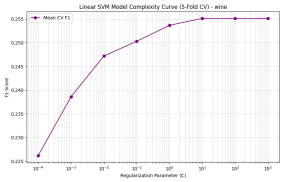
The Decision Tree depicts a rapid increase in mean CV F1 score as the max depth increases from shallow values,



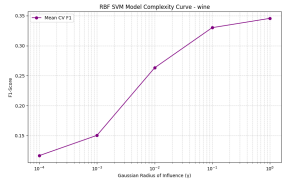
(a) DT ($max\_depth = 7$)



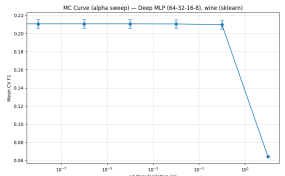
(b) kNN (k)



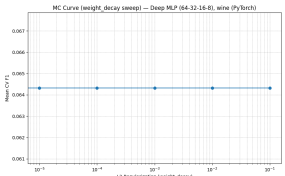
(c) Lin SVM (C)



(d) RBF (γ)



(e) NN (sk)



(f) NN (PT)

Figure 5: Model complexity curves (Wine).

peaking around $max\_depth = 22$ before plateauing (bias is reduced with moderate variance). This indicates that the model captures key decision boundaries earlier on in the dataset, with deeper trees providing little benefit (with the risk of overfitting).

The kNN curve depicts the highest performance at smaller values of k and declines as the number of neighbors increases. This reflects the bias-variance tradeoff, where small k values enable the model to capture local structure, with larger k values introducing higher bias (due to the neighborhoods becoming too large). The downward trend indicates that overly large neighborhoods lower the sensitivity to class boundaries for the model.

The Linear SVM model shows steady improvement in mean CV F1 performance as the regularization increases (C), eventually stabilizing at larger values. This suggests that softer margins initially reduce overall bias. Beyond the optimal margin and misclassification threshold, further increases provide diminishing returns. The RBF SVM also demonstrates increasing performance as γ increases from small values, indicating that low γ promote underfitting due to overly smooth decision boundaries. Performance improves gradually at moderate γ values, suggesting nonlinear relationships are present.

The sklearn MLP shows relatively stable performance across a wide range of regularization values, indicating robustness to small changes in the model's capacity. Extremely high regularization values (> 0) show a drop in performance, reflecting underfitting in a constrained model setup. The PyTorch shows extremely minimal sensitivity to weight decay across the range of values, suggesting the architecture chosen dominates performance. Overall the model complexity curves portray how the Wine dataset is well-structure and can be effectively modeled with moderate complexity.

3.2.2 Learning Curves

Learning curves for the Wine dataset show smoother convergence and smaller gaps between training F1 performance and validation F1 performance, compared to the Adult dataset (Wine dataset is smaller and less complex). Most models reach stable performance relatively quickly, indicating additional training data provides limited improvement once the core patterns are discerned. The same hyperparameter settings selected for the Adult dataset from the model complexity analysis, were also used here to maintain consistent model capacity across datasets.

The Decision Tree exhibits higher training performance at smaller training subset sizes and declines once more data is fed, while validation performance improves modestly before stabilizing. This suggests that the model initially overfits on smaller training subset sizes, and gradually reduces variance as more data is fed to the model. The small gap at larger training subset sizes indicates a balanced bias and variance at $max\_depth = 7$.

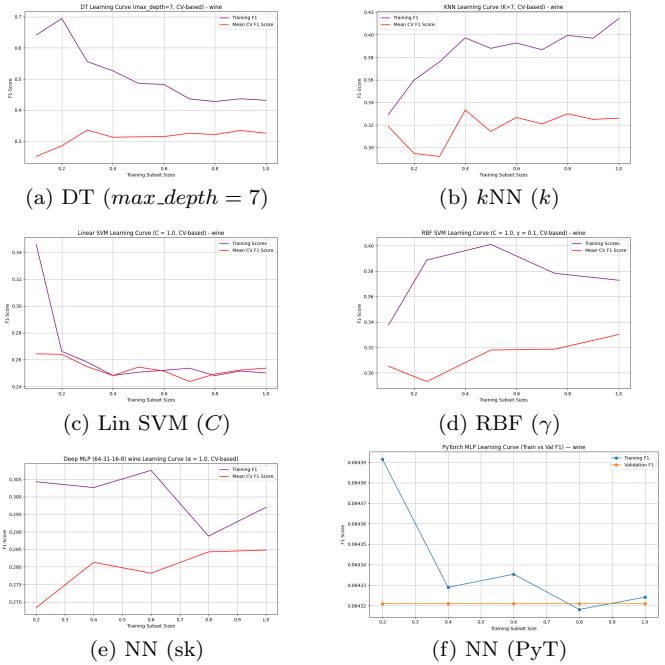
For kNN, training performance increases with more data fed, whereas validation performance remains stable with minor perturbations. The consistent gaps is indicative of moderate variance due to model's sensitivity to its neighborhood. Generalization improves towards larger training subset sizes as well.

The Linear SVM portrays poorer performance regardless of training subset size. The closeness of training and validation curves suggest low variance and relatively higher bias. Performance stabilizes early on, indicating the model already establishes the dominant linear structure of the data with only a subset of the original data. The RBF SVM shows strong training performance with a large gap to the validation performance at smaller training subset sizes. This gap narrows as more data is fed, reflecting the model's flexibility of nonlinear decision boundaries. The model is indicative of overfitting early on, but generalizes more when more data is fed.

Both neural network implementations portray stable learning throughout training subset sizes. The PyTorch MLP shows nearly overlapping training and validation curves, indicating low variance and potentially strong regularization or insufficient capacity for this dataset. The sklearn MLP demonstrates slight improvements in validation performance as more data is fed, suggesting controlled model capacity and overall stable generalization. Overall, the learning curves indicate the Wine dataset can be effectively modeled with subsets of its original size, with most models achieving stable performance without the entire dataset.

3.2.3 Timing Curves

Timing on the Wine dataset reflects its small size, with most models training very quickly as the respective hyperparameters are increased. Decision Trees show increasing fit time with depth as expected, while kNN portrays higher prediction costs due to larger distance computations. The Linear SVM remains computationally efficient with its inference time being much lower than its training time, due to its optimization process in finding the optimal hyperplane boundary. The RBF SVM shows higher inference time due to the increase in support vectors, subsequently making predictions slower than training on smaller datasets. Both Neural Networks take roughly the same amount of time to fit the data, with the sklearn MLP taking the longest training time but reasonable prediction speed. Overall, computational differences in speed are less pronounced for both training and inference when compared to the Adult dataset, due to the reduced dataset size.

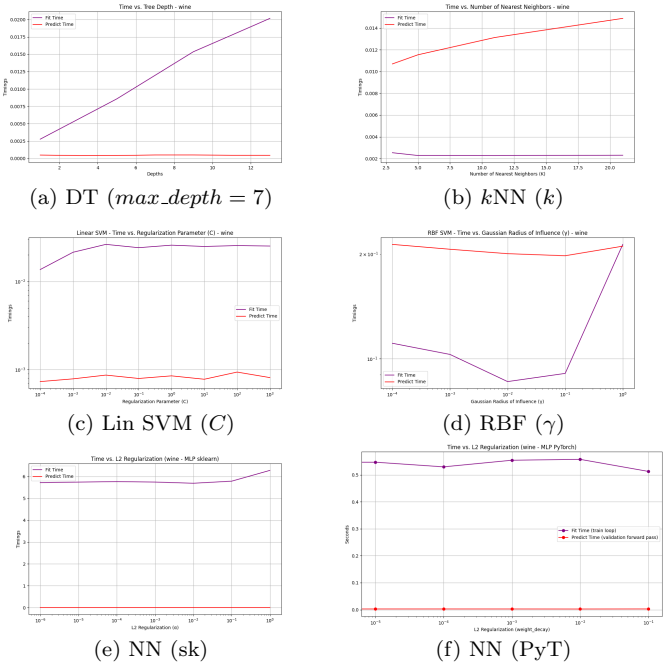


(a) DT ($max\_depth = 7$) (b) kNN (k)

(c) Lin SVM (C) (d) RBF (γ)

(e) NN (sk) (f) NN (PyT)

Figure 6: Learning curves for all models the Wine dataset. **Purple** = Training F1 score, **Red** = Mean CV F1 score.



(a) DT ($max\_depth = 7$) (b) kNN (k)

(c) Lin SVM (C) (d) RBF (γ)

(e) NN (sk) (f) NN (PyT)

Figure 7: Timing curves for all models the Wine dataset. **Purple** = fit time, **Red** = predict time.

3.2.4 Confusion Matrix

For all models, confusion matrices reveal clear differences in class separation ability in a multiclass dataset across models. Most models show a concentration of predictions around the central classes while occasionally struggling with minority classes.

The Decision Tree ($max\_depth = 7$) demonstrates moderate classification performance. There is a strong prediction concentration around class 4, which appears to be the most dominant class. Although instances of classes 3, 4, 5 are correctly classified, there is noticeable confusion with neighboring classes (classes 3-5, 5-6). For example, predicting class 4 when the true label is class 3. This suggest that the tree captures major decision boundaries but struggles with distinctions between adjacent classes. This is indicative of moderate variance and limited model granularity at this depth.

The kNN model shows a similar pattern to the Decision Tree, with strong prediction performance on central classes. However, there is a noticeable spread of misclassifications in neighboring classes (classes 3-5, 4-6) which reflects the local distance-based design of the model (higher bias is shown, tending towards nearby classes).

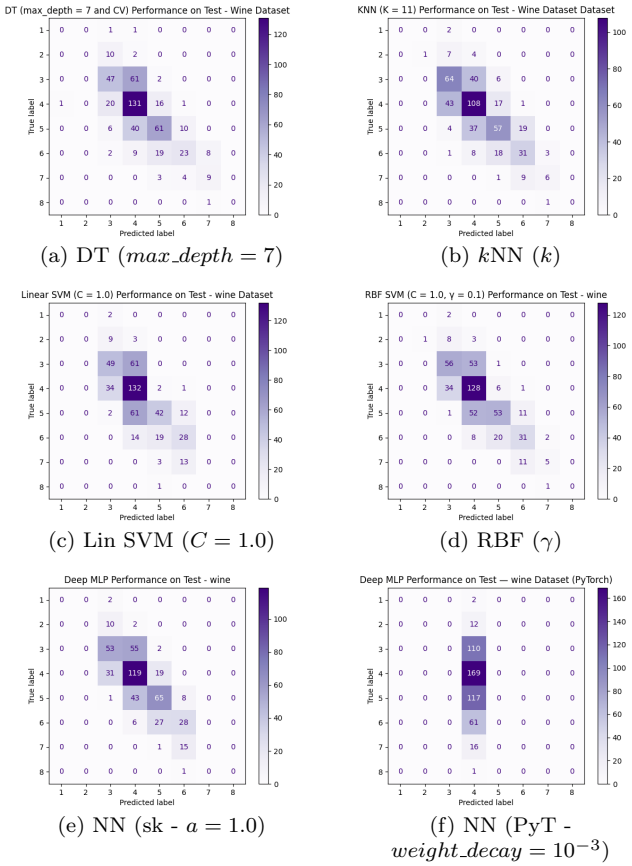
The Linear SVM shows slightly improved classification performance along the diagonal especially for class 4, indicating better separation of the dominant class. There still is some misclassification in neighboring classes (classes 3-5, 4-6) suggesting the linear boundary is insufficient to separate nonlinear relationships in the data. The RBF SVM also shows similar performance, with improved diagonal performance. Similar to its linear counterpart, there is some misclassification along the same neighboring classes. The nonlinear kernel captures more complex features, leading to more balanced performance across the classes.

Similar to the previous models, the sklearn Deep MLP shows reasonable performance with predictions along the diagonal, especially for class 4. However, it is still susceptible to misclassification in the same neighboring classes. This suggests the model is learning meaningful nonlinear patterns but may be under-regularized or limited by the capacity of the dataset. On the contrary, the PyTorch Deep MLP shows minority class prediction collapse, where all predictions are centrally located along class 4. This means the model failed to learn class boundaries to separate on, and converged to predicting the majority class to cut its losses. This is likely due to the architecture of the model being overly complex (deep) for a small dataset, thus resulting in poor generalization.

3.3 Neural Network Analysis

3.3.1 Capacity Scaling – Wide vs. Deep

After testing both wide and deep network architectures for the MLP implementations in both sklearn and PyTorch, the deeper architecture (64-32-16-8) slightly outperforms the wide architecture with a single large hidden layer (90). This indicates that increasing depth in a MLP allows the network to better learn the hierarchy of feature representations in a complex dataset. This is shown and beneficial for the Adult dataset, where the interactions between categorical and numerical features are complex. With the performance differences being modest for the Adult dataset, it's likely that the diminishing returns do not warrant the larger computational cost. For the Wine dataset, the wide architecture completely outperforms the deeper architecture. This suggests that the smaller dataset size does not provide enough data to effectively train a deeper network on, resulting in the lower validation performance. This highlights the importance of how dataset size influences which network architecture is better.



(a) DT ($max\_depth = 7$) (b) kNN (k)

(c) Lin SVM ($C = 1.0$) (d) RBF (γ)

(e) NN (sk - $a = 1.0$) (f) NN (PyT - $weight\_decay = 10^{-3}$)

Figure 8: Confusion matrices for all models the Wine dataset.

3.3.2 Epoch Curves

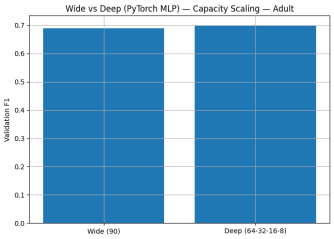
The epoch curve for the Adult dataset depict that training loss decreases steadily throughout training, which points to stable optimization under SGD. Validation loss initially increases before stabilizing, which is indicative of mild overfitting in later epochs. The Wine dataset shows decreasing training loss with validation loss remaining relatively unchanged. This indicates the model continues to fit the training data well and improvements do not translate to better generalization. This lack of meaningful gain in validation is likely due to the smaller dataset size relative to the model's capacity.

Overall, the capacity scaling and epoch curves for both datasets depict how neural network performance is heav-

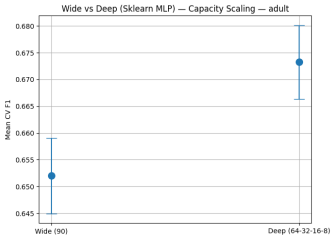
Model	Adult		Wine	
	Acc	F1	Acc	F1
Dummy Baseline	0.752	0.000	0.346	0.064
Decision Tree	0.850	0.651	0.555	0.333
kNN	0.837	0.655	0.547	**0.350**
SVM (linear)	0.845	0.659	0.514	0.252
SVM (RBF)	0.852	0.671	0.561	0.350
MLP (sklearn)	0.850	0.678	0.543	0.272
MLP (PyTorch)	0.820	**0.685**	0.346	0.064

Table 1: Final model comparison on the Adult and Wine datasets against a dummy baseline. A dummy baseline classifier was implemented and evaluated on both datasets.
Final hyperparameters: Decision Tree ($max\_depth = 7$), kNN ($k = 11$), Linear SVM ($C = 1.0$), RBF SVM ($\gamma = 0.1$), and Neural Networks with hidden layers $(64, 32, 16, 8)$.

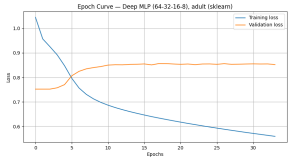
ily reliant on the size of the dataset and the network architecture. Finding a balance between the two is important in designing models that will generalize well. Larger datasets like the Adult dataset benefit from deeper, more complex network architectures. Smaller datasets like the Wine dataset, favor simpler or wider network architectures.



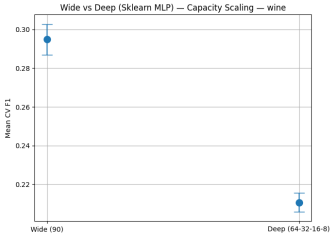
(a) Wide vs. Deep - Adult dataset (PyTorch)



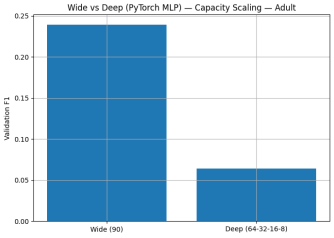
(b) Wide vs. Deep - Adult (sklearn)



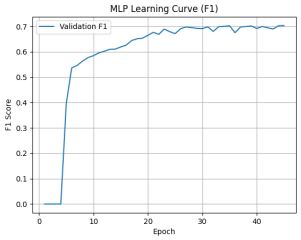
(c) Epoch Curve - Deep MLP - sklearn



(d) Wide vs. Deep - Wine (sklearn)



(e) Wide vs. Deep - Adult (PyTorch)



(f) Epoch Curve - Adult (PyTorch) - Validation F1 Score

Figure 9: Neural Network Analysis of Capacity Scaling (Wide vs. Deep) and Epoch Curves for Adult and Wine datasets

4 Cross-Model Comparison

Across both datasets, clear trade-offs between model capacity, validation performance, and sensitivity to hyperparameter adjustments were clear. On the Adult dataset, high-capacity models like Neural Networks and the RBF SVM portrayed high validation performance due to their ability to capture trends in large-scale nonlinear and complex datasets. The downside of this was increased training time and high-sensitivity to hyperparameter adjustments. Decision Trees also performed well at moderate depths but began to overfit at deeper levels. kNN also showed stable performance, reflecting its sensitivity to the high-dimensional feature space. On the Wine dataset, performance differences were relatively similar due to the low-dimensionality of the data. Simple models like Decision Trees and kNN performed similarly, while the Linear SVM underfit due to the nonlinearity of the data. Computationally, Decision Trees and Linear SVMs were the most efficient, and Neural Networks and RBF SVMs inflicted higher costs with diminishing returns at higher capacities. These findings are indicative of the importance of data preprocessing, and methodical hyperparameter tunings to mitigate the bias-variance tradeoff across models.

5 Conclusion

This study evaluated various supervised learning algorithms on two inherently different datasets. It helped portray how model behavior changes in regard to data complexity and hyperparameter adjustments. The Adult dataset depicted a clear tradeoff due to the class imbalance between Type I and Type II errors. As mentioned before, high-capacity models identified the minority class more often while simpler models favored the majority class more often. The Wine dataset depicted a more even spread of errors with model misclassifications being concentrated along similar classes, due to overlapping feature distributions. Overall, this study reinforced the idea that large datasets better support complex machine learning models, while simpler datasets require more straightforward machine learning models to effectively perform. Future work can explore this study on a different genre of machine learning models, like ensemble machine learning methods that rely on boosting (like XGBoost) or bagging (bootstrap aggregation).

6 References

- Plaut, David C., and Geoffrey E. Hinton. "Learning Sets of Filters Using Back-Propagation." Computer Speech & Language, vol. 2, no. 1, Mar. 1987, pp. 35–61, https://doi.org/10.1016/0885-2308(87)90026-x.
- Cortes, C., & Vapnik, V. (1995). Support-vector networks. Machine Learning, 20(3), 273–297
- Tam, Adrian. "Building Multilayer Perceptron Models in PyTorch - MachineLearningMastery.com." MachineLearningMastery.com, 26 Jan. 2023, machinelearningmastery.com/building-multilayer-perceptron-models-in-pytorch/.
- Zhang, Xinhe. "Multi-Layer Perceptron (MLP) in PyTorch." Deep Learning Study Notes, 26 Dec. 2019, medium.com/deep-learning-study-notes/multi-layer-perceptron-mlp-in-pytorch-21ea46d50e62.
- GeeksforGeeks. "Classification Using Sklearn Multilayer Perceptron." GeeksforGeeks, 11 Oct. 2023, www.geeksforgeeks.org/machine-learning/classification-using-sklearn-multi-layer-perceptron/.
- GeeksforGeeks. "RBF SVM Parameters in Scikit Learn." GeeksforGeeks, 31 July 2023, www.geeksforgeeks.org/python/rbf-svm-parameters-in-scikit-learn/.
- GeeksforGeeks. "F1 Score in Machine Learning." GeeksforGeeks, 27 Dec. 2023, www.geeksforgeeks.org/machine-learning/f1-score-in-machine-learning/.
- GeeksforGeeks. "One Hot Encoding in Machine Learning." GeeksforGeeks, 12 June 2019, www.geeksforgeeks.org/machine-learning/ml-one-hot-encoding/.
- GeeksforGeeks. "Overfitting in Decision Tree Models." GeeksforGeeks, 2 May 2024, www.geeksforgeeks.org/machine-learning/overfitting-in-decision-tree-models/.
- Scikit Learn Documentation (http://scikit-learn.org/)
- PyTorch Documentation (https://docs.pytorch.org/docs/stable/index.html)