A Comprehensive Analysis of Supervised Learning Algorithm Performance on Customer and Spotify Datasets

Shahruk Saeed

February 13, 2025

1 Introduction

This report presents an exploratory analysis of three supervised learning algorithms: Neural Networks, Support Vector Machines (SVM), and k-Nearest Neighbors (KNN). Specifically, it examines how these algorithms perform in two classification problems with distinct characteristics. The first problem involves predicting whether a customer will use a coupon in the sixth month based on a customer personality dataset. This dataset presents challenges such as class imbalance, as there are relatively few instances of coupon usage. Additionally, the dataset contains highly correlated features, making it a valuable test case for evaluating the robustness of machine learning models against these common real-world issues. The second problem uses a Spotify dataset to predict whether someone will dance to a song. This classification task is based on a target variable derived from the "danceability" feature. Unlike the first problem, this dataset is smaller and has balanced classes, providing an opportunity to analyze how data size impacts model performance, particularly for neural networks, which are prone to overfitting on smaller datasets. I hypothesize that for the first dataset, SVM will perform best due to its robustness against class imbalances and highly correlated features. For the second dataset, I expect SVM to outperform both neural networks and KNN, as it tends to generalize better with limited data. To ensure accurate model evaluation, the customer dataset will be cleaned by removing missing or incorrect values and standardizing all features to prevent noise and inconsistencies caused by different feature scales. Similarly, the Spotify dataset's target variable will be derived from the "danceability" feature to create a balanced classification task.

2 Methodology

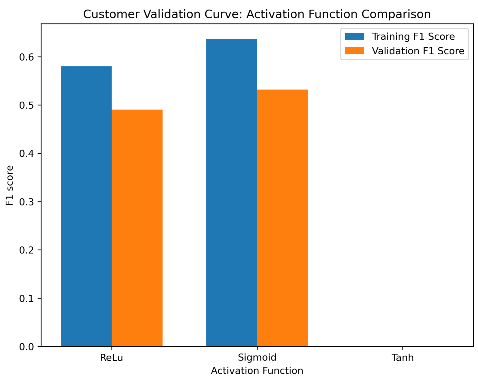
To test my hypotheses, I systematically experimented with Neural Networks, Support Vector Machines, and k-Nearest Neighbors across two distinct datasets. my primary objective was to evaluate how different hyperparameters affect each model's performance, particularly focusing on layer size, depth, and activation functions for NNs, kernel types for SVM, and the value of k for kNN. By exploring these aspects, I aimed to validate my assumption that SVM would outperform the other models, especially in handling class imbalance and correlated features.Each model was tested under multiple configurations to determine the optimal hyperparameters. Neural networks offer high flexibility in capturing complex relationships, but they are susceptible to overfitting and require careful hyperparameter tuning. According to Goodfellow et al. (2016) [4], deep neural networks excel at learning complex relationships but require extensive tuning of hyperparameters such as depth, width, and activation functions to generalize well. To evaluate NN performance, I tested different layer sizes, network depths, and activation functions. I experimented with small (32, 16, 2), medium (64, 32, 16, 2), and large (128, 64, 32, 16, 2) hidden layer sizes. Larger networks were expected to capture more complex relationships but could overfit, especially with smaller datasets (LeCun, Bengio, Hinton, 2015) [6]. I also varied the network depth, testing configurations with one, two, and three hidden layers. Deeper networks were expected to improve performance for complex data but might require more training data to avoid overfitting (Hastie, Tibshirani, Friedman, 2009) [5]. Additionally, I tested three activation functions: ReLU, Sigmoid, and Tanh. I trained models for up to 500 epochs with early stopping to prevent overfitting and used stochastic gradient descent (SGD) as the optimizer with cross-entropy loss. SVMs are effective for high-dimensional, correlated data and can handle class imbalance with proper kernel selection. My key focus was on kernel type, where I tested linear, polynomial, radial basis function (RBF), and sigmoid kernels. Vapnik (1998) [9] introduced SVMs as a robust classification technique that finds an optimal hyperplane to maximize margin between classes. The linear kernel is best suited for linearly separable data, while the polynomial kernel introduces non-linearity with degree tuning. The RBF kernel, which is widely used for non-linearly separable data, was expected to perform best (Schölkopf,Smola, 2002) [**?**]. The sigmoid kernel was tested for completeness but is less commonly used in practical applications. The RBF kernel was expected to outperform others on both datasets, as it provides flexible decision boundaries. The linear kernel was expected to perform well on the customer dataset, given its correlated numerical features. To optimize performance, I tuned the regularization parameter C and the gamma parameter for non-linear kernels. The F1-score was used for the customer dataset, while accuracy was used for the Spotify dataset. kNN is a simple and interpretable model but suffers from sensitivity to feature scaling and class imbalance. The primary hyperparameter of interest was k, the number of neighbors. I tested a range of k values from 1 to 20. A small k value, such as 1 or 3, was expected to result in high variance and likely overfitting. A moderate k value, such as 5 to 9, was expected to provide optimal performance (Mitchell, 1997) [7]. A large k value, such as 15 to 20, was expected to introduce high bias and likely underfitting. The F1-score was used as the primary metric for the customer dataset, while accuracy was used for the Spotify dataset. To ensure fair comparisons, I standardized numerical features, as kNN is highly sensitive to feature scaling (Hastie et al., 2009) [5].To analyze and compare model performance, I employed several evaluation techniques. Learning curves were plotted to observe bias-variance trade-offs, where overfitting was expected in neural networks, while SVM was anticipated to show a smoother generalization curve (Goodfellow et al., 2016) [4]. Hyperparameter tuning involved cross-validation, where I plotted validation curves to assess the impact of specific hyperparameters. For NNs, I analyzed hidden layer size, depth, and activation functions. For SVM, I examined kernel type, and for kNN, I evaluated different k values. Final model evaluation included confusion matrices, which provided insights into false positives and false negatives, particularly valuable for the imbalanced customer dataset.Based on theoretical ML knowledge, several key expectations guided my analysis. SVM with an RBF kernel was expected to outperform other models due to its ability to generalize well with limited data and handle imbalance effectively (Schölkopf, Smola, 2002) [?]. Neural networks were expected to struggle on the Spotify dataset due to its smaller size, leading to overfitting (LeCun et al., 2015) [6]. kNN was expected to be highly sensitive to feature scaling and k-value, potentially underperforming on imbalanced datasets (Hastie et al., 2009) [5].

3 Results

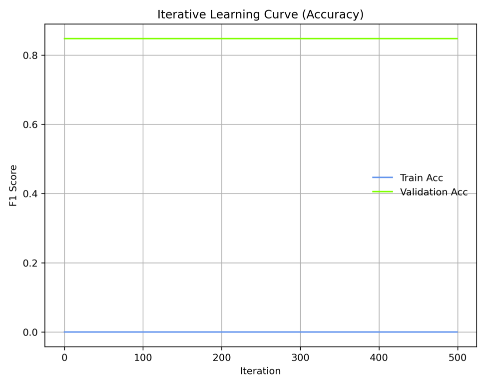
After training and evaluating the three algorithms neural network, Support Vector Machine and K-nearest Neighbors. I analyzed the results using learning curves validation curves and confusion matrices.

3.1 Neural network

Neural networks primarily struggled with overfitting. For the Customer dataset, when analyzing the validation curves I found that reducing the complexity improved the performance of the model and the best performing parameters were a single hidden layer with 16 neurons and used the Sigmoid activation function. The final model achieved an F1-score of 0.42, the parameter with the best performance with all other values equal was sigmoid. When looking at the learning curves for the customer dataset below we can see that as training size increased, training accuracy remained significantly higher than validation accuracy and constant which signify's that the model was over fitting highly and only predicting one class leading to a validation score of 0. this indicates that the Neural Network greatly struggled with the class imbalance and could have benefited from dropout.On the Spotify dataset, Neural Networks showed strong performance in training and in validation meaning the model generalized well. The best architecture, with two hidden layers containing 32 and 16 neurons, achieved a final test accuracy of 0.90. In this case I see that the neural network handled working with a smaller dataset better than imbalanced data and this may because there were sufficient examples of each of the target class in the spotify dataset. This shows the assumption in the second hypothesis was wrong in terms of the model struggling to fit the smaller dataset.

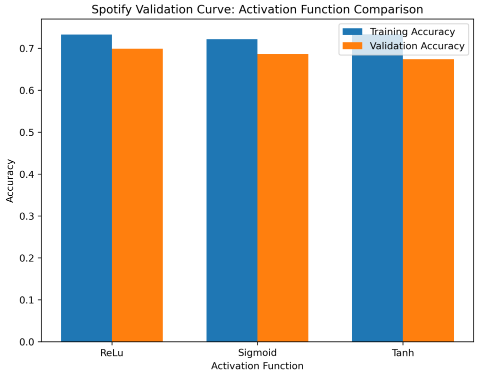


(a) Activation Function Analysis

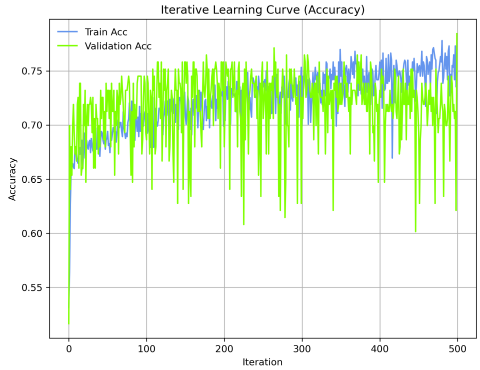


(b) Customer NN Learning Curve

Figure 1: Customer NN Performance Analysis

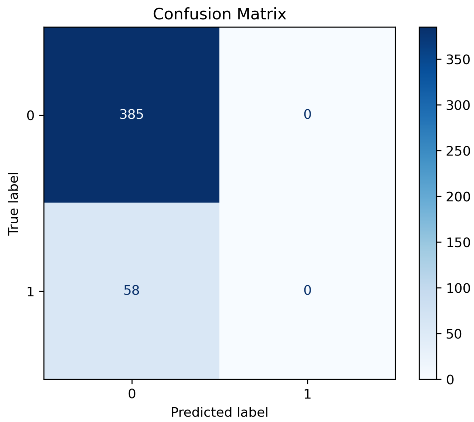


(a) Activation Function Analysis

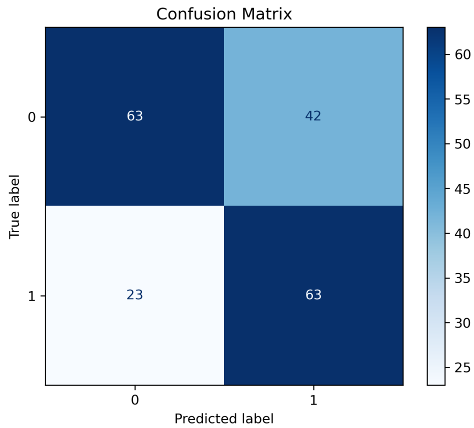


(b) Spotify NN Learning Curve

Figure 2: Spotify NN Performance Analysis



(a) Customer NN Confusion Matrix

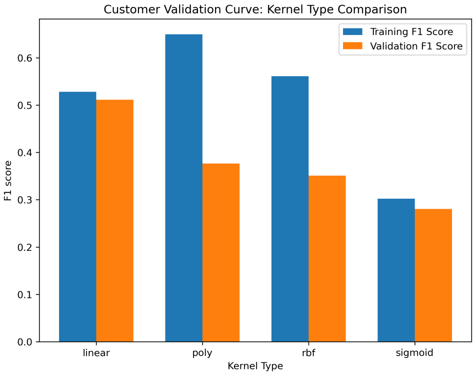


(b) Spotify NN Confusion Matrix

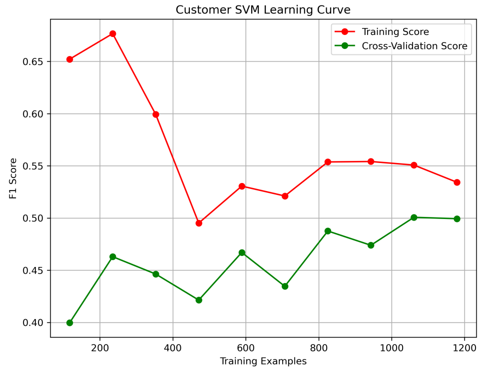
Figure 3: NN Confusion Matrix

3.2 Support Vector Machine

Support Vector Machines performed the best among the three models on both datasets. On the Customer dataset, it achieved an F1-score of 0.893. The final model used a linear kernel. The confusion matrix showed that the model maintained a strong balance between precision and recall, performing significantly better than the Neural Network and k-Nearest Neighbors. This performance indicates that Support Vector Machines effectively handled correlated features and class imbalance. By maximizing the margin between classes, SVMs reduce overfitting and improve generalization, particularly in high-dimensional datasets [1].On the Spotify dataset, Support Vector Machines achieved the highest accuracy of 0.707. The final model used a polynomial kernel, which performed well in capturing complex decision boundaries. The confusion matrix showed improved class separation compared to other models, suggesting that SVMs successfully extracted relevant features from the data. The polynomial kernel's effectiveness implies that the dataset's decision boundary was non-linear, requiring a transformation to achieve better separability. Prior research indicates that kernel-based SVMs perform well when data is not linearly separable [8], which aligns with this result.

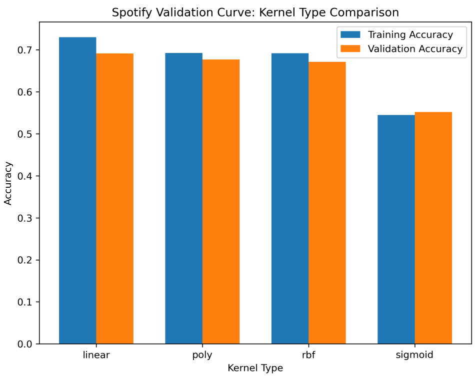


(a) SVM Kernel function

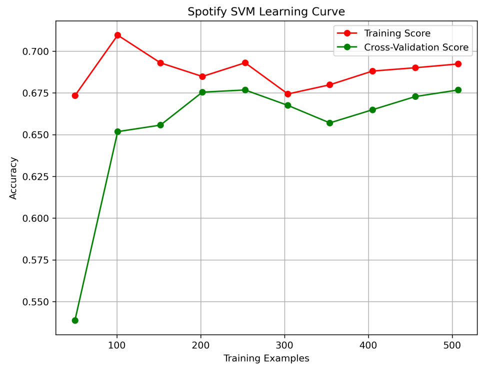


(b) Customer SVM Learning Curve

Figure 4: Customer SVM Performance Analysis

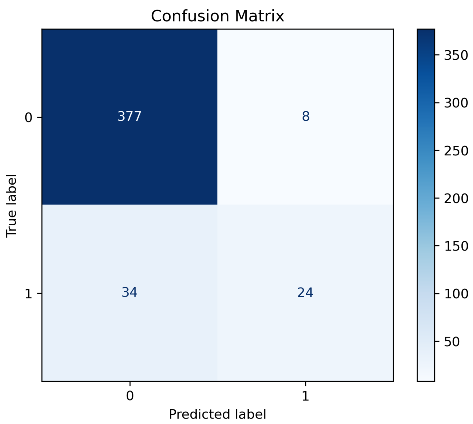


(a) SVM Kernel function

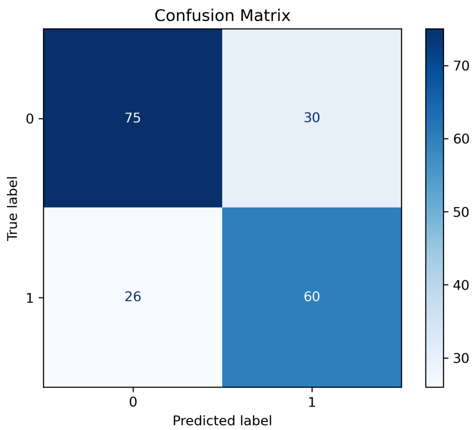


(b) Spotify SVM Learning Curve

Figure 5: Spotify SVM Performance Analysis



(a) Customer SVM Confusion Matrix

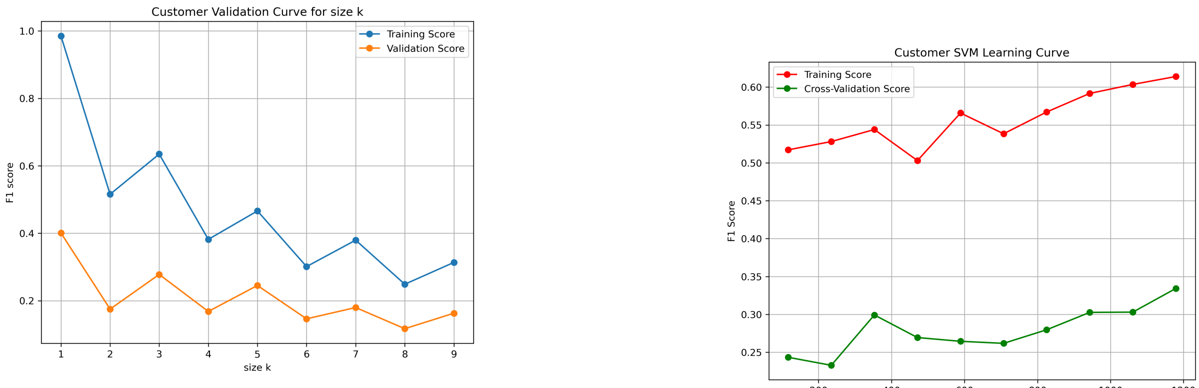


(b) Spotify SVM Confusion Matrix

Figure 6: SVM Confusion Matrix

3.3 k-Nearest Neighbors

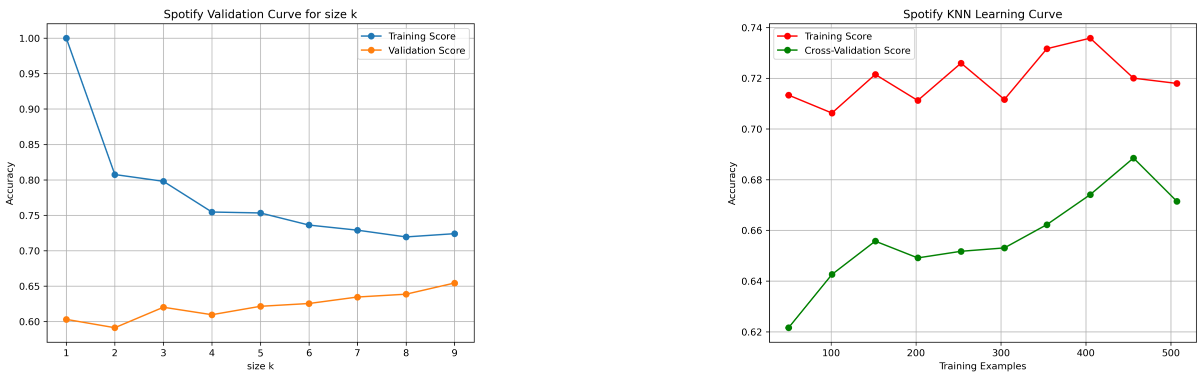
k-Nearest Neighbors demonstrated moderate performance on the Customer dataset, achieving an F1-score of 0.841. The final model used $k = 3$. The confusion matrix indicated that it had difficulty classifying the minority class, though it performed slightly better than the Neural Network. The model's sensitivity to feature scaling and class imbalance resulted in inconsistent classification boundaries. Since k-Nearest Neighbors relies on distance-based similarity, variations in feature scaling and sparsity in the minority class affected its ability to make accurate predictions [2]. The poor performance in this setting suggests that kNN is not well suited for datasets with high feature correlation and imbalanced classes. On the Spotify dataset, k-Nearest Neighbors achieved an accuracy of 0.681. The final model used $k = 9$. The confusion matrix showed a moderate misclassification rate, suggesting that while kNN performed better than the Neural Network, it still struggled to define clear decision boundaries. The model's reliance on local patterns meant that the limited dataset size restricted its ability to generalize beyond training samples. Increasing k helped smooth decision boundaries but also reduced sensitivity to finer details in the data, leading to a tradeoff between reducing variance and maintaining class separability. Prior studies show that kNN often struggles with small datasets, as it depends on local density rather than learning a broader decision function [3].



(a) Customer K Value Validation Curve

(b) Customer KNN Learning Curve

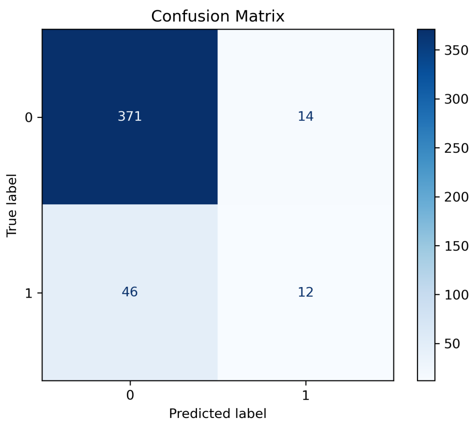
Figure 7: Customer KNN Performance Analysis



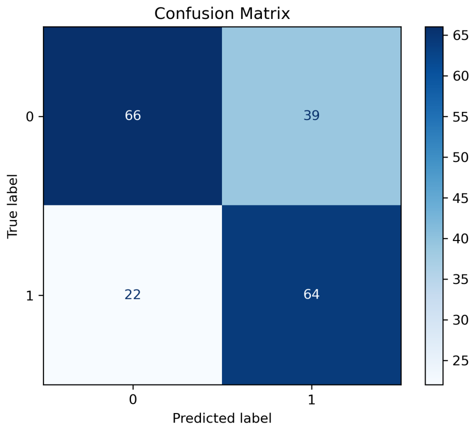
(a) Spotify K Value Validation Curve

(b) Spotify KNN Learning Curve

Figure 8: Spotify KNN Performance Analysis



(a) Customer KNN Confusion Matrix



(b) Spotify KNN Confusion Matrix

Figure 9: KNN Confusion Matrix

3.4 Model Comparative analysis

The results support the hypothesis that Support Vector Machines would perform best, as it effectively handled both correlated features and imbalanced classes in the Customer dataset and achieved the highest accuracy on the Spotify dataset. Its ability to generalize across datasets highlights its robustness. Neural Networks underperformed as expected in the Spotify dataset due to its smaller size, reinforcing the importance of large datasets for deep learning models. k-Nearest Neighbors struggled the most, validating the hypothesis that its performance would be limited by class imbalance and feature correlation.Neural Networks could be improved by implementing class-weighted loss functions or oversampling techniques to mitigate class imbalance. Expanding the dataset, particularly for the Spotify dataset, could enhance the model's ability to generalize. Experimenting with alternative activation functions and deeper architectures could also improve feature extraction and classification accuracy.Support. Vector Machines performed well, but further optimization could include tuning hyperparameters like the regularization parameter (C) or exploring different kernel functions. The success of the polynomial kernel on the Spotify dataset suggests that testing higher-degree kernels or radial basis function (RBF) kernels might yield even better results. For k-Nearest Neighbors, performance could be improved by applying proper feature scaling, employing dimensionality reduction techniques like Principal Component Analysis (PCA) to mitigate the impact of correlated features, and using distance-weighted voting to refine classification decisions. Optimizing k through cross-validation could also help fine-tune the model's sensitivity and bias-variance tradeoff.

4 Conclusion

These findings align with the original hypotheses. The assumption that Support Vector Machines would outperform the other models was confirmed, as it demonstrated the best generalization ability across both datasets. Neural Networks, as expected, struggled on the smaller, imbalanced Spotify dataset, reinforcing the need for large datasets when using deep learning models. k-Nearest Neighbors performed inconsistently, particularly in handling feature correlation and class imbalance, which was anticipated given its reliance on local data structures.Overall, dataset characteristics played a crucial role in determining model performance. Support Vector Machines proved to be the most reliable choice for datasets with correlated features and class imbalance. Neural Networks showed potential but require more data to be effective. k-Nearest Neighbors struggled due to its sensitivity to feature scaling and dataset size. Future research could explore ensemble methods, such as boosting, to further improve classification performance. Additionally, experimenting with hybrid models that combine SVM and deep learning approaches may provide better adaptability across different dataset types.

References

[1] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.

[2] Thomas Cover and Peter Hart. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13(1):21–27, 1967.

[3] Evelyn Fix and Joseph L. Hodges. Discriminatory analysis: Nonparametric discrimination: Consistency properties. 1951.

[4] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016.

[5] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Springer, 2009.

[6] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

[7] Tom M. Mitchell. *Machine Learning*. McGraw-Hill, 1997.

[8] Bernhard Schölkopf and Alexander J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, 2002.

[9] Vladimir Vapnik. *Statistical Learning Theory*. Wiley, 1998.