# CS7641 Assignment 1
# Supervised Learning
# Spring 2025

Chau Nguyen

## I. Introduction

Given 2 entirely different datasets – Consumer Personality Analysis and Most Streamed Spotify Songs 2023 – this paper will explore two distinct machine learning problems: a classification task predicting the customer's income class and a regression task estimating the number of Spotify streams of a song. By leveraging different supervised learning models, including Neural Networks, Support Vector Machines (SVM) and k-Nearest Neighbors (kNN), this aims to uncover patterns in consumer behavior and music streaming trends based on relevant features.

The first problem involves classifying customers into three income levels – low, mid, or high – based on their demographics, spending behaviors, and engagement metrics. Understanding these income groups can help businesses tailor marketing strategies and customer interactions more effectively. Given the structured nature of the dataset (customer demographics, spending behavior, and engagement features), if the data exhibits linear separability among income groups, we expect Support Vector Machines (SVM) to outperform Neural Networks (NN) and k-Nearest Neighbors (kNN) in classification due to its ability to maximize the margin between classes.

The second problem focuses on predicting a song's popularity on Spotify based on its performance on other streaming platforms, release date, and audio features, such as danceability, valence, and key. Given that streaming counts are likely influenced by nonlinear relationships (e.g., trends, release date effects, and audio features), Neural Networks (NN) are expected to perform best, as they can capture complex patterns in the data.

Through this exploration, we seek to gain insights into how different supervised learning techniques handle the underlying nature of these datasets – structured consumer data with categorical and numerical features versus time-sensitive, trend-driven music streaming data with complex audio features. By comparing model performances, we aim to determine the most suitable techniques for each problem. This analysis will help inform future model selection by identifying which algorithms best capture the patterns within each dataset and generalize effectively to unseen data.

## II. Dataset Exploration & Preprocessing

### A. Consumer Personality Analysis Dataset

The Consumer Personality Analysis dataset [**?**] provides a detailed analysis of a company's customers, aiming to help businesses better understand their target audience and optimize products and marketing strategies [**?**]. It includes various demographic attributes (such as age, education level, and marital status), spending behaviors (such as total spending across different product categories), and engagement metrics (such as frequency of store purchases and response to promotions).

*1) Income-Level Classification Problem:* For this dataset, the classification task involves predicting a customer's income level – categorized as low, mid, or high – based on these features. The assumption is that demographic and spending behaviors correlate with income level, allowing for effective separation of classes in feature space. If these relationships are well-structured, we expect the income levels to be linearly separable or nearly separable with an appropriate feature transformation. Given this assumption, SVM should perform well by maximizing the margin between classes. However, if the relationships are more complex and nonlinear (as engagement metrics might not directly correlate to income), neural networks may be better suited to capturing intricate patterns.

This classification task is particularly interesting from a machine learning perspective because it presents a balance between structured patterns and potential complexity, making it a suitable problem for comparing models with different inductive biases. Linear classifiers like SVM can be evaluated against nonlinear approaches like neural networks, while instance-based methods like kNN provide an alternative perspective on the structure of the feature space. Additionally, the class imbalance in income levels introduces challenges related to model evaluation, requiring careful selection of performance metrics beyond simple accuracy.

*2) Feature Selection:* Since the consumer incomes are provided as integers ranging from \$1,730 to \$666,666 in the dataset, we transformed this into a classification problem by categorizing income into three discrete bins:

- Low-income: \$0 to \$40,000 (Label 0)
- Mid-income: \$40,000 to \$80,000 (Label 1)
- High-income: \$80,000 to \$666,666 (Label 2)

This binning resulted in an imbalanced distribution, where most data points fall within the middle-income range, followed by the low-income range, while the high-income category contains significantly fewer samples.

An alternative approach to defining income categories would be to use equal-sized bins based on the 33rd and 66th percentiles of the income column, ensuring a more

balanced class distribution. However, our chosen binning strategy better reflects the realistic unequal distribution of wealth in the United States, where the middle income class forms the majority, followed by lower-income, while high-income is the minority.

To visualize the imbalance, Figure 1 illustrates the income class distribution in our dataset compared to the unequal wealth distribution in America. This imbalancce poses a challenge for classification models, as they may struggle to learn meaningful patterns in less-distributed classes, or the high-income class. Thus, instead of using accuracy as the performance metric for the models' evaluation, we will use F1 score to ensure a balanced assessment of preceision and recall across the three income classes. The F1 score is useful in this context because it mitigates the impact of class imbalance by considering both false positives and false negatives, preventing the model from being biases toward the majority class.

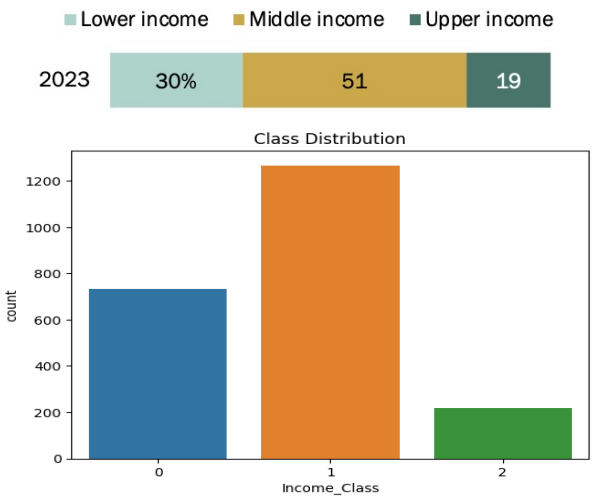


Fig. 1. Income Level Distribution in United States vs. dataset

*3) Data Preprocessing:* We have transformed the data columns and selected a total of 13 relevant data attributes: age, education level, marital status, children, customer tenure (days since joining), recency (number of days since last purchase), spending, web purchases, catalog purchases, store purchases, website visits, deal purchases, and engagement. Engagement is a new feature that has 3 classes, based on how fast customer respond to promotional campaigns, aiming to capture the underlying relationship between customer responsiveness and income level.

We also performed several pre-processing steps, including encoding categorical variables (including education, marital status, and engagement) and normalizing numerical features. The dataset is split into 80% training and 20% test data, and both input features and output labels are standardized.

### B. Most Streamed Spotify Songs 2023 Dataset

The Most Streamed Spotify Songs 2023 dataset [**?**] contains a list of the most popular songs of 2023 on Spotify. It includes various attributes such as song name, artist, release date, total streams across multiple music streaming sites, and audio features (e.g., danceability, valence, and energy).

*1) Spotify Streams Prediction Problem:* For this dataset, the task involves predicting a song's total Spotify stream count based on its release date, popularity on other platforms, and audio features, formulating a regression problem. Popularity trends on other music platforms like Apple Music and the frequency of a song appearance in playlists and charts may have a linear relationship with Spotify streams, suggesting that SVM and kNN could perform well. However, audio features such as danceability, valence, and tempo may exhibit nonlinear relationships with streaming success, making neural networks better suited for capturing complex patterns.

This problem is interesting from a machine learning perspective because it requires models to balance structured numerical features (e.g., tempo, loudness) with context-dependent factors (e.g., external platform popularity). The task allows for direct comparisons between linear models like SVM, instance-based methods like kNN, and deep learning approaches like NNs, evaluating their ability to generalize to unseen music trends. Additionally, since this is a regression problem, it's important to consider multiple error metrics to have a comprehensive understanding of the model's performance. Metrics used are Mean Squared Error (MSE), Mean Absolute Error (MAE), and R-squared. They help capture different aspects of prediction accuracy, providing a more hollistic view of how well the models predict Spotify stream counts.

*2) Feature Selection & Data Preprocessing:* For feature selection, track name and artist(s) name are excluded to ensure generalization and focus on the other 22 more relevant numerical and categorical features. Categorical features like key and mode were hot-encoded, and any missing key values were replaced with the median. Numerical features were standardized by replacing non-numeric values with the column mean, followed by normalization to have zero mean and unit variance. After one-hot encoding, the dataset consists of 31 input features. The dataset was split into 80% for training and 20% for testing with standardized features.

## III. Income-Level Classification Problem

### A. Hyperparameter Tuning

*1) Neural Network:* Hyperparameter tuning was performed using a grid search approach to optimize the neural network model for income-level classification. The search explored different configurations for the neural network's architecture, including the layer sizes and activation functions. Specifically, we tested two layer configurations: (16, 64, 3) and (16, 32, 3), along with two activation functions: Tanh and Sigmoid.

Prior to performing the grid search, manual tuning was conducted to find the optimal learning rate. A learning rate of 0.001 was found to provide the most stable training in 500 epochs. Additionally, we observed that using the ReLU activation function led to weak validation accuracy,

suggesting it may not capture the underlying patterns in the data effectively or there exist large outliers in the dataset. Experiments with different network architectures revealed that a single hidden layer yielded the best results. Deeper and more complex models, such as a hidden layer with 128 neurons, or two hidden layers, led to significantly worse and unstable training performance. This could be attributed to the relatively small size of the dataset, where more complex models are prone to overfitting. A dropout rate of 0.1 was used to regularize training since a larger value significantly impacted performance.

The model was wrapped in Skorch's NeuralNetClassifier, with key parameters including the Adam optimizer, CrossEntropyLoss criterion, and a dropout rate of 0.1. A 5-fold cross-validation was used for model evaluation.

The grid search identified the best hyperparameters based on cross-validation performance, with the optimal model architecture consists of 16 input neurons, 64 hidden neurons, 3 output neurons, and using Tanh activation function (Figure 2). The relatively small architecture prevents overfitting, while Tanh helps capture non-linear relationships without the vanishing gradient problem, leading to better convergence and improved generalization compared to other configurations.

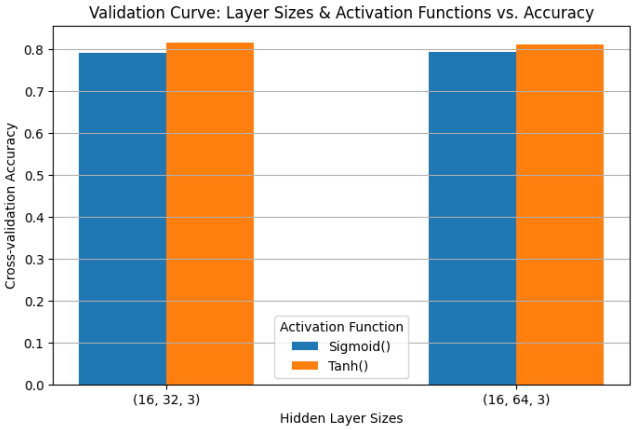


Fig. 2. Validation Curve for NN Model (Consumer dataset)

*2) Support Vector Machine (SVM):* For the SVM model, hyperparameter tuning is performed on different kernel functions: linear, radial basis function (RBF), polynomial, and sigmoid. The goal was to identify the kernel that best suited the data by evaluating the model's performance using the F1 score, which accounts for class imbalances in the target variable.

For each kernel type, a SVM model is trained and the resulting weighted F1 score on the test set is stored for comparison. The linear kernel resulted in the highest F1 score (Figure 3), likely because the relationship between income class and consumer demographics, spending behaviors, and engagement is relatively linear, making the linear kernel more effective at separating the classes compared to more complex kernels like RBF and polynomial. Thus it is chosen as the final model for performance evaluation in the next section.

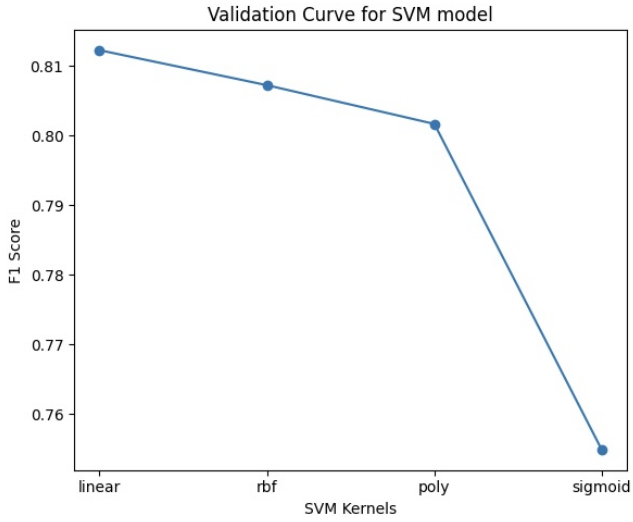


Fig. 3. Validation Curve for SVM Model (Consumer dataset)

*3) k-Nearest Neighbor (kNN):* For the kNN mdoel, hyperparameter tuning was performed by testing different k values from 1 to 25 to identify the optimal number of neighbors for the classification task. The model's performance was evaluated using the weighted F1 score to handle class imbalance.

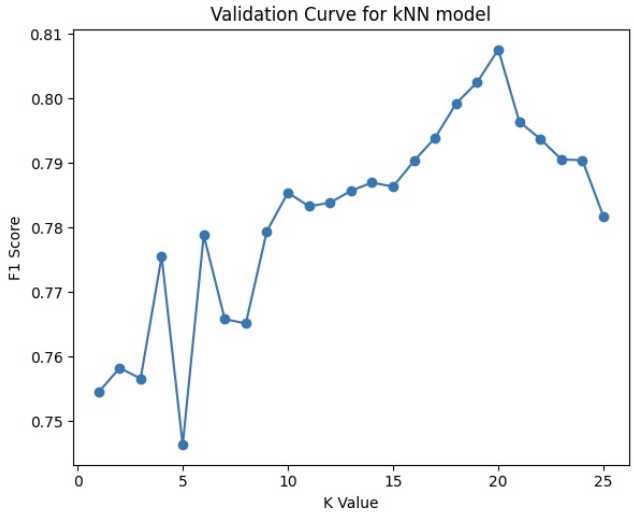


Fig. 4. Validation Curve for kNN Model (Consumer dataset)

The validation curve (Figure 4) reveals that low k values (from 1 to 15) results in F1 scores below 0.8, indicating that the model is overly sensitive to outliers in the data, leading to low bias, high variance, and poor generalization to unseen data. On the other hand, larger k values dilute the influence of any single neighbor and might smooth out important patterns, leading to higher bias and lower F1 score.

### B. Training Performances

*1) Neural Network:* The learning curve (Figure 5), which plots training loss against validation loss, along with the accuracy curve (Figure 6) showing train and validation accuracy, indicates signs of overfitting. Specifically, the training loss is around 67% while the validation loss is slightly higher at 73%. The training accuracy is relatively high at 89%, but the validation accuracy is only around 82%.

This discrepancy suggests that while the model is performing well on the training data, it struggles to generalize effectively to the test set. The model may have memorized the training data too well, capturing noise or specific patterns that do not generalize well to unseen data.

In addition, the fact that the validation loss and accuracy become stagnant after around 70 epochs suggests that the model has converged. Further training no longer improves performance, likely due to overfitting or memorizing noise in the small training set. At this stage, the model has likely exhausted its ability to improve generalization on the validation set, and further epochs may not provide meaningful gains. Early stopping could be a potential strategy to prevent overfitting and save computational resources.

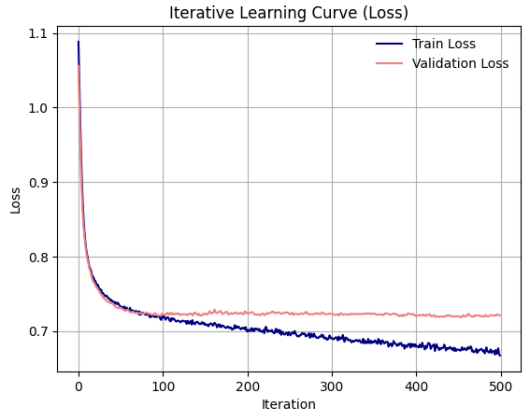


Fig. 5. Learning Curve (Loss) for NN Model (Consumer dataset)

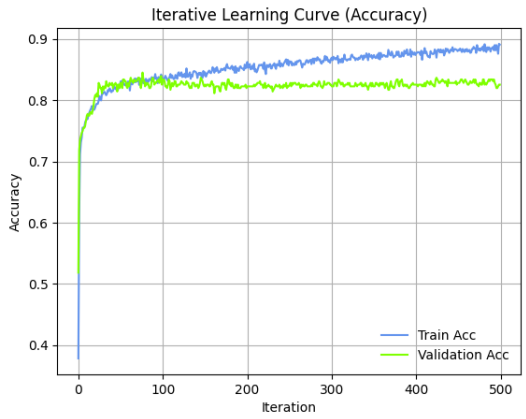


Fig. 6. Accuracy Curve (Loss) for NN Model (Consumer dataset)

In fact, if we reduce the number of epochs to 70, we get the following learning curve (Figure 7), where validation loss is lower than training loss. This suggests that to prevent overfitting beyond this point, additional data, such as more training data or data augmentation, is required to maintain model improvement and avoid stagnation.

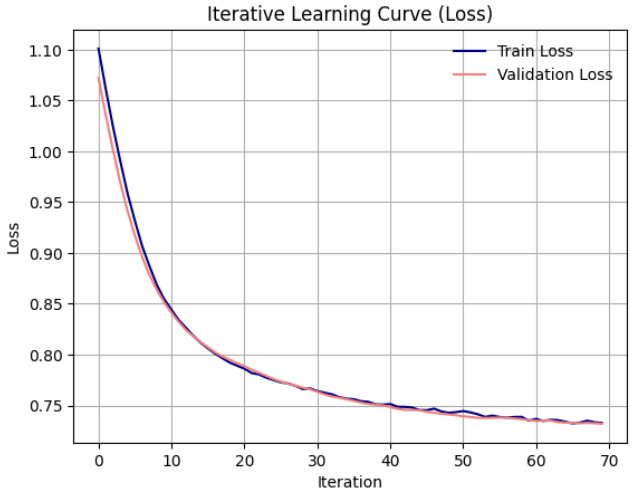


Fig. 7. Learning Curve (Loss) for NN Model with Early Stopping

*2) Support Vector Machine (SVM):* The learning curves for SVM and kNN models are generated by using the learning_curve function from sklearn, which computes the performance of the model across different training sizes (from 10% to 100%) using 5-fold cross-validation. The mean and standard deviation of the scores are calculated and shaded to represent the variability in performance.

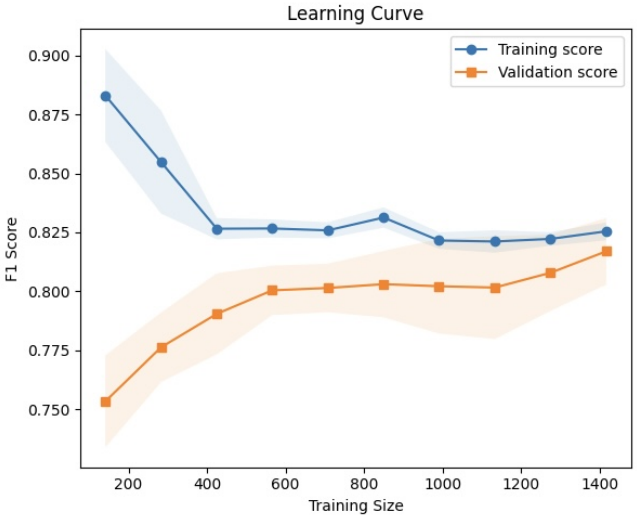


Fig. 8. Learning Curve (F1 Score) for SVM Model (Consumer dataset)

The learning curve for the SVM model (Figure 8) shows that both the training and validation scores improve as the training size increases, but they never intersect, maintaining a noticeable gap. The training curve moves towards the center, reflecting the model's stabilizing performance as it learns. In

SVMs, this can happen if the model is not overfitting and stabilizes as it processes more data. The validation curve increases as more of the training set is learned, suggesting that the model continues to generalize better with more data. The persistent gap indicates the model is not overfitting and is effectively learning general patterns. Moreover, the validation score's continued increase at the end of the training set implies that additional data could further improve the model's performance.

*3) k-Nearest Neighbors (kNN):* The learning curve for the kNN model shows that the training score consistently increases with more data, remaining significantly higher than the validation score. This suggests that the model may be overfitting, as it continues to perform better on the training set compared to the validation set, even with additional data. The persistent gap between the curves indicates the model is memorizing training examples rather than generalizing well to unseen data, as well as struggling with imbalanced data.

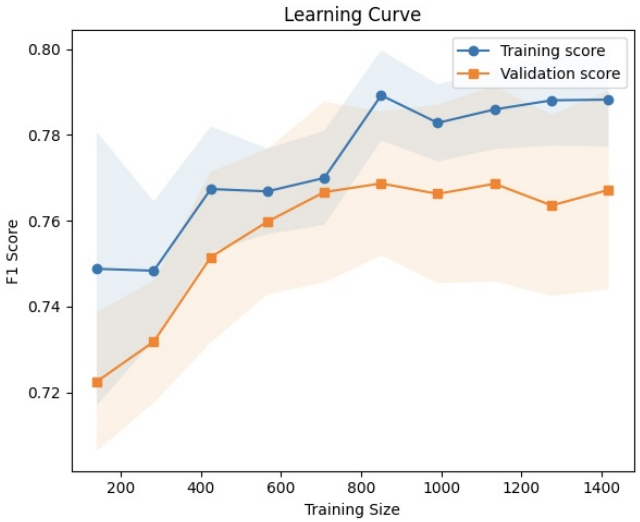


Fig. 9. Learning Curve (F1 Score) for kNN Model (Consumer dataset)

### C. Results Analysis

The performances of the three models can be summarized as follows:

- The neural network model achieved the highest F1 score of 0.8270 but exhibited clear signs of overfitting, with validation performance stagnating after 70 epochs. This indicates that while the model is capable of capturing complex relationships among consumer demographics, spending behaviors, and engagement, it memorizes the training data due to limited training samples.
- The SVM model, particularly with the linear kernel, demonstrated a steadily improving validation curve with increasing training size, while maintaining a consistent gap with the training curve. This behavior indicates that the income classes are reasonably well-separated by a hyperplane in the feature space and the underlying relationships are predominantly linear.
- The kNN model showed a persistent gap with higher training scores, underscoring its difficulty in handling the high-dimensional, imbalanced nature of the dataset and effectively capturing nuanced relationships between income levels and the varied consumer features.

In the data pre-processing stage, approximately 30 rows containing NaN entries were dropped to ensure data integrity, while appropriate encoding and normalization were applied. However, this reduction in data may have contributed to the non-converged learning curve observed for the SVM model. Had these NaN values been transformed – using techniques similar to those applied to the Spotify dataset – more data could have been retained for training, potentially leading to better convergence and improved classification performance for the SVM model.

| Neural Network | SVM | kNN |
|---|---|---|
| 0.8270 | 0.8123 | 0.8076 |

TABLE I
CLASSIFICATION PERFORMANCE: F1 SCORES ON TEST SET

Table 1 summarizes the F1 scores for the three models on the test set. As shown, the neural network achieved the highest F1 score (0.8270) but also exhibited signs of overfitting. In contrast, the linear SVM attained a competitive F1 score of 0.8123 with a much simpler architecture and more stable performance, suggesting that additional or augmented training data could further boost its performance. Meanwhile, the kNN model scored 0.8076, indicating that it struggles to capture the complex relationships in this imbalanced dataset.

Therefore, the SVM model with a linear kernel is chosen as the best model for income-level classification. Its performance (second-highest F1 score of 0.8123), characterized by balanced training and validation scores, indicates that the relationship between consumer demographics, spending behaviors, and engagement is predominantly linear, allowing the model to generalize well. Additionally, SVMs are inherently scalable and have robust performance on imbalanced datasets—an important consideration given the unequal distribution of income levels. These factors together suggest that the linear SVM is the most effective and appropriate choice for predicting income levels in this dataset.

## IV. SPOTIFY STREAMS REGRESSION PROBLEM

### A. Hyperparameter Tuning

*1) Neural Network:* The hyperparameter tuning process for the neural network regression model is evaluated across three model architectures (showing only hidden layers) – (32, 64, 32), (64, 128, 64), and (128, 64, 128) – combined with three activation functions (ReLU, Tanh, and Sigmoid). Each model was trained with 100 epochs with a fixed learning rate of 0.001 and a dropout rate of 0.3, using mean squared error as the loss function. The model performance was assessed based on the validation loss, and the configuration yielding the lowest loss was selected as the best.

Preliminary manual tuning revealed that a learning rate of 1e-3 provided stable convergence, whereas lower rates converged too slowly and higher rates led to overfitting. A dropout layer is added after every hidden layer to regularize training; thus, dropout of 0.3 was chosen because larger dropout values significantly degraded performance.

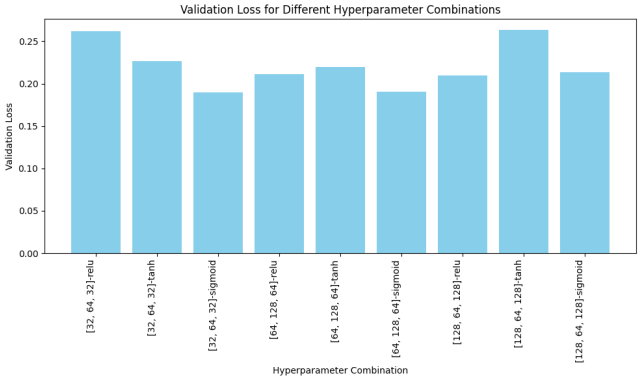


Fig. 10. Validation Curve for NN Model (Spotify dataset)

The best performance was achieved with the (64, 128, 64) architecture using Sigmoid activation (Figure 10). This configuration outperformed the simpler (32, 64, 32) model, which likely lacked sufficient capacity to capture the complexity of the relationships between among audio features, cross-platform stream counts, and chart performances, and the more complex (128, 64, 128) model, which tended to overfit given the limited size of the dataset. The superior performance of Sigmoid activation may be attributed to its bounded output, which helps stabilize learning and avoid issues such as saturation and dead neurons that can occur with ReLU and Tanh.

*2) Support Vector Machine (SVM):* Hyperparameter tuning was performed using Support Vector Regression (SVR) with four different kernel functions: linear, RBF, polynomial, and sigmoid. The goal was to identify the kernel that best captures the underlying relationship between the song's audio features, cross-platform popularity, chart performances, and release information, by minimizing the Mean Squared Error (MSE) on the test set. For each kernel, an SVR model was trained on the training data and its performance evaluated on the test set, with the resulting MSE values recorded for comparison and selecting the kernel yielding the lowest MSE.

The results indicate that the RBF kernel achieves the lowest MSE (0.2), followed by the linear kernel, then the polynomial kernel (0.3), with the sigmoid kernel performing the worst (1.8). The RBF kernel's superior performance suggests that the relationship between the song's audio features and Spotify streams is highly nonlinear. Its ability to map data into a higher-dimensional space allows it to capture these complex patterns more effectively. The linear kernel performs moderately well, indicating that a substantial linear component exists in the data (i.e., Spotify chart performances and streams), but it lacks the flexibility to fully model the

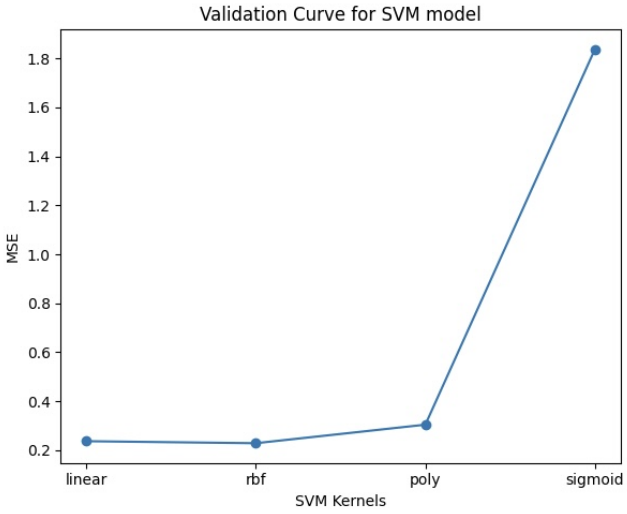


Fig. 11. Validation Curve for SVM Model (Spotify dataset)

nonlinearity. The polynomial kernel introduces some nonlinearity, but it may be overly complex. Finally, the sigmoid kernel's poor performance is likely due to its sensitivity to hyperparameter tuning and its less effective mapping for this particular regression task, resulting in a significantly higher error.

*3) k-Nearest Neighbors (kNN):* The hyperparameter tuning process for kNN regression involved varying the number of neighbors k values from 1 to 25 and evaluating each model's performance using the Mean Squared Error (MSE) on the test set. For each k, the model was trained on the training data and then used to predict the test set, with the MSE computed for each configuration. The k value of 3 resulted in the lowest MSE of around 0.38 was selected as the optimal parameter, balancing bias and variance for this regression task.

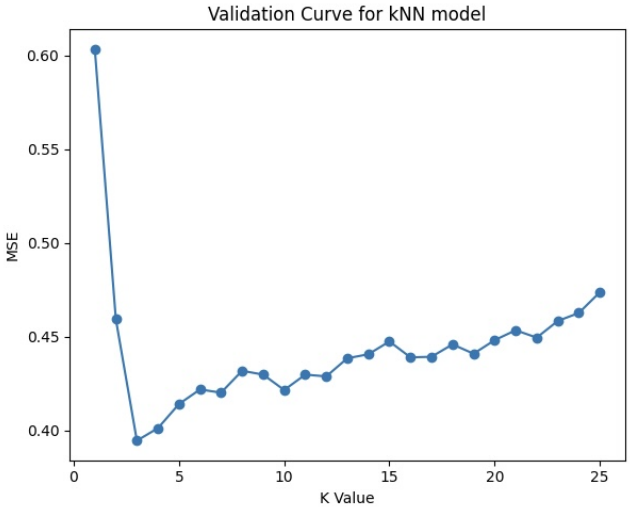


Fig. 12. Validation Curve for kNN Model (Spotify dataset)

In depth, k value of 1 has the highest MSE of 0.61. This

results in a very flexible model with low bias because it can fit the training data very closely. However, it is extremely sensitive to noise in the training data, leading to high variance. As k value increases to 2 and 3, model averages over more neighbors, reducing variance because influence from noisy data points is diminished. The bias slightly increases because the model becomes less sensitive to local fluctuations. As k becomes larger, the model starts averaging over many neighbors, which can lead to a significant increase in bias. Here, the model underfits the data, meaning it cannot capture the underlying patterns effectively.

### B. Training Performances

*1) Neural Network:* The learning curve for the neural network (Figure 13), based on mean squared error (MSE), shows that the model generalizes well—its validation loss is lower than the training loss. This suggests that the network is successfully capturing the underlying patterns in the data. Neural networks excel at modeling complex, nonlinear relationships in high-dimensional feature spaces. In this case, the network is able to balance the more straightforward, linear associations – such as those between Spotify chart performances and streaming counts – with the complex nonlinearity among the high-dimensional audio features. By leveraging multiple hidden layers architecture and nonlinear activation function Sigmoid, the neural network effectively learns both the broad trends and subtle nuances in the data, leading to robust predictions of Spotify stream counts.

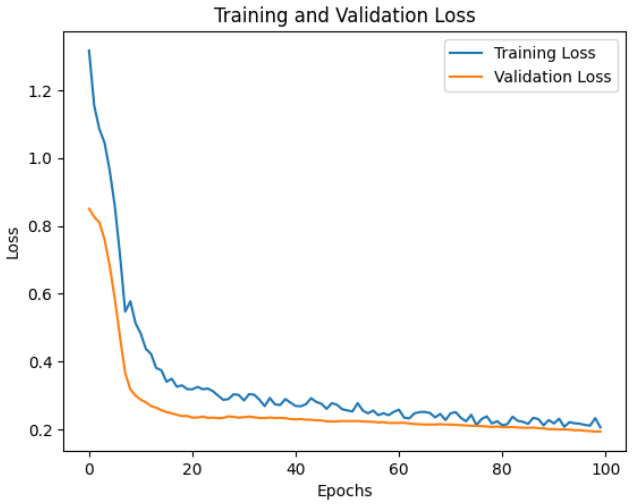


Fig. 13. Learning Curve (Loss) for NN Model (Spotify dataset)

*2) Support Vector Machine (SVM):* The learning curve for the SVM model (Figure 14) reveals a significant gap between the training and validation losses. The training loss decreases from around 0.33 to 0.12, indicating that the model is fitting the training data quite well. However, the validation loss only decreases from approximately 0.51 to 0.33, suggesting that the model is not generalizing as effectively to unseen data. This gap implies that the SVM is struggling to capture the underlying nonlinear relationships present in the dataset. In particular, while the SVM is able to optimize its decision boundary for the training data, its capacity to model complex interactions among features – such as those found in high-dimensional audio features and cross-platform streaming metrics – is limited. As a result, the SVM's inability to adequately separate the classes based on these nonlinear patterns leads to relatively higher validation loss compared to the training loss.

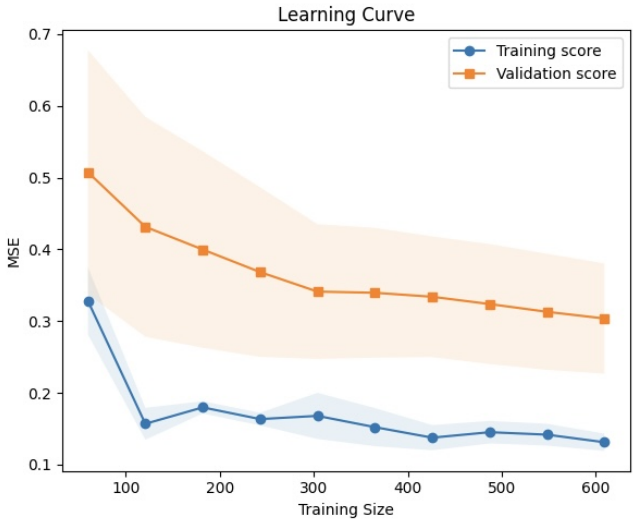


Fig. 14. Learning Curve (Loss) for SVM Model (Spotify dataset)

*3) k-Nearest Neighbors (kNN):* The learning curve for the kNN model (Figure 15) demonstrates the poorest performance among the evaluated models. While the training loss decreases slightly from 0.38 to 0.28 as the more training data is processed, the validation loss remains significantly higher – dropping only from 0.75 to 0.6. This persistent huge gap indicates that kNN is struggling to generalize to unseen data. Its reliance on distance metrics in high-dimensional spaces, coupled with sensitivity to class imbalance, means that kNN fails to capture the underlying complex relationships effectively. Consequently, the kNN model exhibits inferior performance compared to the neural network and SVM models.

### C. Results Analysis

Among the models evaluated for predicting Spotify stream counts, the neural network achieved the best overall performance. According to Table 2, the neural network model results in the lowest MSE of 0.1945 and the lowest MAE of 0.3072, combined with the highest $R^2$ score of 0.7696, indicate that it effectively captures the complex nonlinear relationships between audio features, cross-platform streaming metrics, and release data. In contrast, the SVM model yielded an MSE of 0.2281, an MAE of 0.3363, and an $R^2$ score of 0.7298, suggesting that while it is competitive, its ability to model intricate nonlinear interactions is somewhat limited compared to the neural network. The kNN model, with an MSE of 0.3945, an MAE of 0.4406, and an $R^2$ score

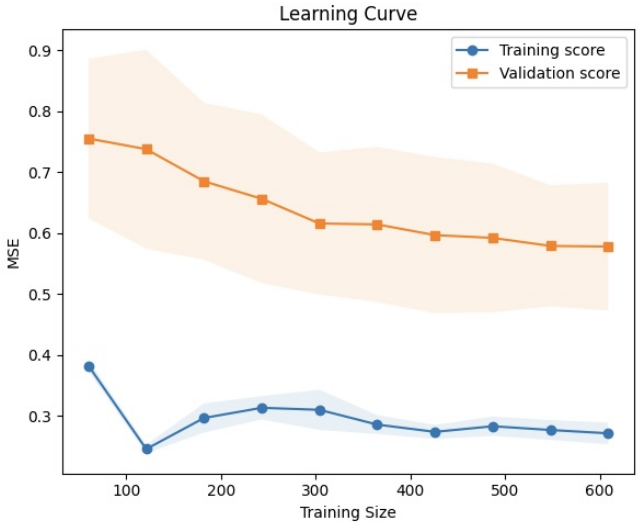


Fig. 15. Learning Curve (Loss) for kNN Model (Spotify dataset)

of 0.5326, performed the poorest, likely due to its sensitivity to high-dimensional feature spaces and imbalanced data.

These differences highlight that the neural network's deep architecture and nonlinear activation functions enable it to better capture the nuanced patterns in the data. Its superior performance indicates that it can more effectively explain the variability in Spotify stream counts, making it the most suitable choice for this regression task when compared to SVM and kNN.

| Model | MSE | MAE | $R^2$ Score |
|---|---|---|---|
| Neural Network | 0.1945 | 0.3072 | 0.7696 |
| SVM | 0.2281 | 0.3363 | 0.7298 |
| kNN | 0.3945 | 0.4406 | 0.5326 |

TABLE II
REGRESSION PERFORMANCE: ERROR METRICS ON TEST SET

We can understand how these different error metrics in Table 2 describe the SVM model's performance:

- Mean Squared Error (MSE) quantifies the average squared difference between predicted and actual values, penalizing larger errors more heavily. In our analysis, a lower MSE indicates that the model produces predictions closer to the actual stream counts, with the neural network achieving the lowest MSE (0.1945).
- Mean Absolute Error (MAE) measures the average absolute differences between predictions and true values, providing an intuitive measure of error in the same units as the target variable. The neural network's lowest MAE (0.3072) confirms its superior accuracy in terms of average prediction error.
- $R^2$ score (coefficient of determination) indicates the proportion of variance in the target variable explained by the model; a higher $R^2$ suggests a better fit. The neural network's highest $R^2$ (0.7696) demonstrates that it captures most of the variability in Spotify stream counts, making it the best performer among the models evaluated.

## V. CONCLUSION

In conclusion, our experiments with supervised learning models reveal that performance varies considerably between the two tasks. For the income-level classification problem, the SVM model with a linear kernel emerged as the best performer, as expected from the hypothesis. Its balanced training and validation scores indicate that it effectively captures the linear relationships among consumer demographics, spending behaviors, and engagement metrics without succumbing to overfitting. Although the neural network achieved high training accuracy, its tendency to overfit – combined with the kNN model's struggles in high-dimensional, imbalanced settings – suggests that the linear SVM is the most suitable supervised learning model for this classification task.

For the Spotify streams regression problem, the neural network model delivered superior performance, as evidenced by its lower error metrics (MSE and MAE) and higher $R^2$ score. This success is attributed to its capacity to capture the complex, nonlinear interactions among audio features, chart performances, and cross-platform streaming metrics. Looking ahead, future work could focus on expanding the datasets through additional training data or data augmentation, as well as refining feature engineering techniques. Further hyperparameter tuning and the exploration of advanced machine learning techniques are expected to enhance model generalization and predictive accuracy in real-world applications.

## REFERENCES

[1] A. Patel, "Consumer Personality Analysis Dataset," 2021. [Online]. Available: https://www.kaggle.com/datasets/imakash3011/customer-personality-analysis/data [Accessed: 02-01-2025].
[2] E. Elgiriyewithana, "Most Streamed Spotify Songs 2023," 2023. [Online]. Available: https://www.kaggle.com/datasets/nelgiriyewithana/top-spotify-songs-2023/data [Accessed: 02-01-2025].
[3] R. Kochhar, "The state of the American Middle Class," Pew Research Center, https://www.pewresearch.org/race-and-ethnicity/2024/05/31/the-state-of-the-american-middle-class/ [Accessed: 02-01-2025].
[4] D. Hamilton, R. Pacheco, B. Myers, and B. Peltzer, "KNN vs. SVM: A comparison of algorithms," US Forest Service Research and Development, https://research.fs.usda.gov/treesearch/62328 [Accessed: 02-01-2025].