Comparative study of Support Vector Regression and Feedforward Neural Network on Spotify Recommendation System

April 7, 2023

1. Introduction

Music is a crucial element of modern life, bringing the enjoyment and vitality to individuals. Spotify, a prominent provider of streaming services, provides subscribers the opportunity to engage with great music and remain current with emerging trends and advancements. The Discover Weekly on Spotify is an automatically generated personalized playlist for users based on their listening habits and preferences. Updated every Monday with 30 songs, it uses algorithms to match users' tastes while introducing them to new artists and genres.

This paper aims to present a comparative study of two machine learning algorithms – the Support Vector Regression (SVR) model and the Feedforward Neural Network (FNN) model – on Spotify recommendation system. This study provides an in-depth understanding of the foundational principles of each model, how both SVR and FNN models are constructed and evaluated using Spotify data. This empirical assessment allows for the observation of their performance and effectiveness in a real-world setting, contributing valuable insights into the application of these models in music recommendation system.

2. Methodologies

2.1. Study Area

Spotify for Developers[1] offers a wide range of possibilities to utilize the extensive catalog of Spotify data, one of which are the audio features calculated for each song and made available via the Spotify Web API[2]. In this study, the author's personal Spotify data will be used. This dataset comprises 945 songs with associated audio features, streaming time and metadata

[1] https://developer.spotify.com/
[2] https://developer.spotify.com/documentation/web-api

such as artist, album, track name and URI. This study will employ content-based filtering method, utilizing SVR and FNN algorithms to explore the efficacy of Spotify recommendation system.

2.2. Support Vector Regression

Support Vector Regression is used to predict a continuous dependent variable (i.e. streaming time of a song) based on a set of independent variables (i.e. audio features of each song). It learns a function that maps the input features to the target output variable.

There are three major hyperparameters for the SVR. *C* is the regularization parameter that controls the trade-off between achieving a low training error and a low testing error. Gamma (γ) is a parameter for non-linear hyperplanes. The higher the γ value, the more likely the algorithm will fit the training data exactly, which may lead to overfitting. A lower γ value will lead to more generalization and underfitting. SVR also introduces a parameter Epsilon (ε), which defines a margin of tolerance for errors. The Kernel Function is related to these three hyperparameters and is used to transform the input data into a higher dimensional space, where it becomes easier to find a hyperplane that separates the data.

The goal of SVR is to find a hyperplane that separates the streaming time values within the specified Epsilon-tube, rather than precisely on the hyperplane itself. The approach allows for some errors within the specified tolerance margin around the hyperplane, which helps create a more robust and generalized model. It facilitates balancing the trade-off between the model complexity and predictions accuracy, while also managing outliers and noise in the data.

The basic mathematical form of SVR is:

$$f(x) = w^T \varphi(x) + b$$

where w is a weight vector, $\varphi(x)$ is the feature mapping function, and b is a bias term. The feature mapping function $\varphi(x)$ transforms the input features x into a higher-dimensional feature space, where a linear decision boundary can be used to separate the data points.

2.3. Feedforward Neural Network

Feedforward Neural Network consists of three types of layers: input layer, hidden layer and output layer. Each layer consists of one or more neurons. The input layer receives the data (i.e. 11 audio features) and passes the data through the hidden layer to produce an output (i.e. streaming time) in the output layer. The hidden layer applies nonlinear transformations to the input data and produces features that will be used by the output layer to generate recommendations.

An active function is applied to the output of each neuron in FNN. Its purpose is to introduce non-linearity into the output of a neuron, allowing the network to learn more complex relationships between input and output. Common and popular active functions include Rectified Linear Unit (ReLU), tanh, and sigmoid.

The basic mathematical form of a neural network model involves a series of matrix multiplications and nonlinear transformations,

$$y = f(w_2 * f(w_1 * x + b_1) + b_2)$$

where x is the input vector, w_1 and w_2 are weight matrices for connections between input layer and the hidden layer, and between the hidden layer and the output layer, b_1 and b_2 are bias vectors for the hidden layer and output layer, f is a nonlinear activation function, and y is the output vector.

2.4. Performance Indicators

Performance indicators are used to evaluate the SVR and FNN to determine which method is appropriate to predict the streaming time based on audio features. Mean Squared Error (MSE) measures the average squared difference between the predicted and actual values. Root Mean Squared Error (RMSE) is the squared root of the MSE and is more interpretable than MSE because it is expressed in the same units as the dependent variable. Normalized RMSE (nRMSE) is a modified version of the RMSE that considers the scale of the target variable. It is calculated by dividing the RMSE by the range of the target variable. NRMSE puts the prediction error on a standardized scale, making it easier to compare the performance of different models. R-squared measures the proportion of the variance in the dependent variable that is explained by the independent variables in the model. It usually ranges from 0 to 1, with higher values indicating a better fit of the model. In this study, RMSE, nRMSE and R-squared are chosen to measure the performance of SVR and FNN models.

2.5. Current state of arts and existing methodologies

While most of the literatures compare different methods within Support Vector Machine or Neural Network models, there are a few studies compared the performance of these two algorithms in the fields such as environmental science, finance, and biotechnology. Achieng (2019) compared SVR and FNN for modeling soil moisture retention curves, and found SVR outperformed FNN in terms of prediction accuracy and robustness to outliers. Rouf et al. (2021) found SVR was the most popular technique used for stock price prediction; however, techniques like neural network models provided more accurate and faster predictions. Other studies have investigated the use of hybrid models that combine SVR and FNN, or other machine learning algorithms. For instance, Chen et al. (2015) proposed a hybrid prediction model of SVR and general regression neural network (GRNN) would be the most suitable

model for the design of the fed-batch fermentation conditions for Iturin A production.

Current studies on Spotify recommendation system poses unique challenges due to the vast array of styles and genres, social influences, and geographic factors that shape listener preferences. The large number of potential recommendations can be reduced by suggesting albums or artists, but this may not suit the system's intended use and may disregard non-homogenous artist repertoires. Two popular methods are used for recommendation system. The first is the content-based filtering method, which uses audio features to predict the streaming time. This approach focuses on the content of the songs to generate recommendations. The second method, collaborative filtering, relies on usage patterns, i.e. the combinations of items that users have consumed or rated, and how the items relate to each other. However, this method encounters the cold start problem, hindering its effectiveness in recommending songs for new users (Naina et al., 2022).

To date, no research has been conducted on the performance of SVR and FNN in the context of the Spotify recommendation system. As content data is readily accessible and can be utilized for the comparative study, this study employs the content-based filtering method to assess the effectiveness of SVR and FNN and compare their respective performance.
To date, no research has been conducted on the performance of SVR and FNN in the context of the Spotify recommendation system. As content data is readily accessible and can be utilized for the comparative study, this study employs the content-based filtering method to assess the effectiveness of SVR and FNN and compare their respective performance.

2.6. Barriers, Issues, and Open Problems

When applying either SVR or FNN for a real-world problem, some concerns may need to consider. With a large number of audio features, the dimensionality of the dataset can become an issue. High dimensionality will increase the complexity of the models, which can lead to overfitting and reduce the generalization performance. To handle this issue, feature selection may be required to reduce the dimensionality while retaining the most significant information. Moreover, FNN is susceptible to overfitting, especially when the dataset is small or noisy. The dataset used in this study consists of 945 observations, which may not be enough to train a complex model like FNN. Additionally, training an FNN can also be computational expensive.

3. Implementation and Comparison

3.1. Study on Support Vector Regression

The dataset was split into train, validation and test sets. The train and validation sets were used to train and tune the model, while the test set was used to evaluate the final model's performance. All the 11 audio features (i.e. danceability, energy, key, loudness, mode, speechiness, acousticness, instrumentalness, liveness, valence, and tempo[3]) were included for SVR to predict the target streaming time (i.e. msPlayed) variable. GridSearchCV[4] was used to perform a hyperparameter search, which tested different combinations of c, γ, and ε values for the Kernel function. The Radial Basis Function (RBF) was chosen in this case because it is a versatile type which works well with a variety of data types and structures with computational efficiency and ease of use. The result showed the model with the best

[3] The audio features of key, loudness, and tempo were transformed to a scale of 0 to 1, to ensure consistency with the scaling of the other audio features.

[4] GridSearchCV is a method in Python scikit-learn that allows to search for the best combination of hyperparameters for a given model using cross-validation. It returns the hyperparameters that give the best performance on the validation data. This method helps automate the hyperparameter tuning process, which saves a lot of time and efforts.

hyperparameters was $c = 100$, $\gamma = 0.01$, and $\varepsilon = 1$, with an RMSE of approximately 5,513,095, an nRMSE of approximately 4.0%, and an R-squared of -0.17.

3.2. Study on Feedforward Neural Network

Same approaches as SVR were followed in terms of the train/validation/test set split and the selection of the input variables used in the model. This ensured consistency in the input data used for the models and allowed for a fair comparison of two models' performances.

Next, the architecture of the FNN was created based on four layers. The input layers were 11 audio features and the output layer was the streaming time. Two hidden layers were chosen in this study to allow the neural network to learn more complex relationships between the input features and the output feature. The number of neurons in each hidden layer, and the range of values used to search for the optimal number of neurons, were hyperparameters to be tuned during the model building process. Three types of active functions, ReLu, tanh and sigmoid, were chosen. The learning rate determined the step size at each iteration while moving toward a minimum of a loss function during the training process. The number of neurons from 32 to 512 in steps of 32 as well as the range of values of 0.01, 0.001, and 0.0001 of the learning rate were chosen in this case because these values are found to be effective through experimentation and optimization during the development of the FNN model.

After that, the best hyperparameters were selected based on the minimum validation loss of the model. The result showed that the model with the best hyperparameters found had four layers, where the first layer was the input layer with 11 audio features inputs, the second and third hidden layers were dense layers with 192 neurons and 64 neurons, and the final output

layer had one neuron. The total number of trainable parameters in the model was 14,721[5], which includes the weights and biases for each layer. The model has an RMSE of approximately 5,253,167, an nRMSE of approximately 3.9%, and an R-squared of -0.07.

3.4. Comparison and Further Analysis

3.4.1. Comparison of SVR and FNN

Table 1: Comparison of SVR and FNN Models

Training Function	Data Set	Model Details	RMSE	nRMSE	R-squared
SVR	Random Split; 64% training data, 16% validation training data, 20% testing data	Kernel function: RBF c = 100 γ = 0.01 ε = 1	5,513,095	4.0%	-0.17
FNN	Random Split; 64% training data, 16% validation training data, 20% testing data	Input layer: 11 input features Number of hidden layers = 2 Neurons of the first hidden layer = 192 Neurons of the second hidden layer = 64 Total parameters = 14,721	5,253,167	3.9%	-0.07

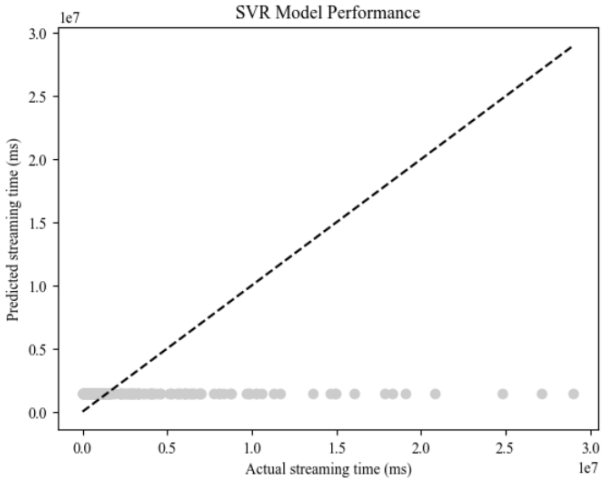
The parallel development of SVR and FNN models were carried out to assess the predictive performance of the models. For this, same inputs were used for the development and comparison of the two algorithms.

The RMSE of SVR was 5,513,095, which indicated on average the predictions were off by 5,513,095 milliseconds (or 92 minutes) from the actual values. This is a relatively small error, considering the range of streaming time is 135,097,816 milliseconds (or 2,252 minutes), which is quite large. The nRMSE of 4.0% was calculated by normalizing the RMSE using the range of streaming time, which allowed for a more interpretable result. The R-squared of -0.17 indicated on average the model did not fit the data well. Nevertheless, a negative R-squared value indicates that the model does not capture the trend in the data and is likely

[5] Input layer: 11 audio features inputs
First dense layer: (11 inputs * 192 neurons) + 192 biases = 2,304
Second dense layer: (192 inputs * 64 neurons) + 64 biases = 12,352
Output layer: (64 inputs * 1 neurons) + 1 bias = 65
Total trainable parameters: 2,304 + 12,352 + 65 = 14,721

overfitting or underfitting the data.

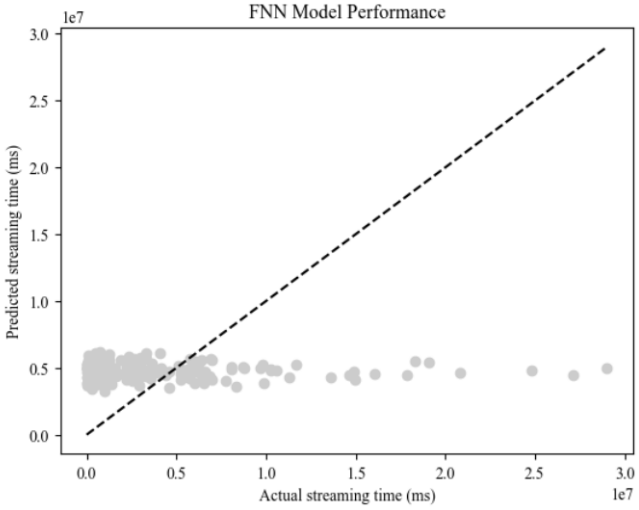
Figure 1: SVR Model Performance



To visualize the model performance, a plot of the predicted values against the actual values was drawn above. This revealed how well the model performed for different ranges of actual values. Additionally, a line of perfect predictions (y=x) was shown in the plot to demonstrate how far the predictions deviate from the actual values. The x-axis and y-axis showed the actual and predicted streaming time respectively. Each point on the scatter plot represented a sample of the split test set. It was shown that the majority of the points were located away from the diagonal line, with their predicted values differing from the actual values. Specifically, the prediction errors, which corresponded to the distances between the predicted and actual values, were mostly below 10,000,000 milliseconds, translating to approximately 167 minutes.

The RMSE of FNN was 5,253,167, which indicated on average the predictions were off by 5,253,167 milliseconds (or 88 minutes) from the actual values. The nRMSE of FNN was 3.9%. Both the RSME and nRMSE of FNN were slightly smaller compared to those of SVR. The R-squared of -0.07 indicated that the FNN performed slightly better than SVR, but overall, still did not fit the data well.

Figure 2: FNN Model Performance



Same as SVR, a plot of the predicted values against the actual values and a line of perfect predictions (y=x) were drawn for FNN. It was shown that the majority of the points were located away from the diagonal line. There were a few points located on the diagonal line, meaning the predicted streaming time was identical to the actual streaming time. Compared to SVR, the data points in FNN were more tightly clustered, meaning a narrower range of prediction errors. The prediction errors, which corresponded to the distances between the predicted and actual values, were mostly below 4,000,000 milliseconds, translating to approximately 67 minutes. This suggested that FNN had a higher level of accuracy in predicting the streaming time based on audio features.

However, the negative R-squared in both models indicated that the chosen models were not a good fit for the data. The models were not able to capture the variability in the data. Given that the RMSEs and nRMSEs for both models were relatively small, it could be valuable to test the model on new, unseen data to validate its performance further.

3.4.2. Further Analysis

In addition to testing SVR and FNN on split test set, this study further validated these two models' performance on new, unseen data. To accomplish this, a new dataset of Global

Weekly Top 100 Songs from Spotify Chart[6] was obtained as the test dataset. This dataset includes 100 top popular songs associated with audio features together with other metadata such as the artist, album, track name, peak rank, source, etc. A new variable, namely pred_msPlayed, was calculated by fitting the audio features data to two models. After that, the top 20 songs with the highest predicted streaming time were selected as recommendations. The author listened these 20 songs recommended by SVR and FNN respectively, rated each song, and determined whether to add them to her library. This approach can provide a comprehensive evaluation of the models' effectiveness on a complete different set of data and may provide additional insights into the models' performances.

Table 2: List of Songs based on SVR Model

Order	Track	Artist	Rating[7]	Added
1	I'm Good (Blue)	David Guetta, Bebe Rexha	6	No
2	Bones	Imagine Dragons	7	Yes
3	ANTIFRAGILE	LE SSERAFIM	5	No
4	Yandel 150	Yandel, Feid	4	No
5	Under The Influence	Chris Brown	6	No
6	LOKERA	Rauw Alejandro, Lyanno, Brray	6	No
7	Shinunoga E-Wa	Fujii Kaze	4	No
8	OMG	NewJeans	5	No
9	Miss You	Oliver Tree, Robin Schulz	7	Yes
10	Bloody Mary	Lady Gaga	4	No
11	PUNTO G	Quevedo	3	No
12	Ditto	NewJeans	8	Yes
13	Escapism. – Sped Up	RAYE, 070 Shake	7	Yes
14	Hype Boy	NewJeans	4	No
15	Superhero	Metro Boomin, Future, Chris Brown	3	No
16	Tití Me Preguntó	Bad Bunny	10	Yes (already in library)
17	As It Was	Harry Styles	5	No
18	Sure Thing	Miguel	7	Yes
19	Rich Flex	Drake, 21 Savage	7	Yes
20	STAR WALKIN'	Lil Nas X	6	No

Table 3: List of Songs based on SVR Model

Order	Track	Artist	Rating	Added
1	Murder In My Mind	Kordhell	9	Yes
2	Calm Down	Rema, Selena Gomez	3	No
3	Sweater Weather	The Neighbourhood	2	No
4	Calm Down	Rema	3	No
5	Made You Look	Meghan Trainor	5	No
6	Leão	Marília Mendonça	1	No
7	Que Vuelvas	Carin Leon, Grupo Frontera	4	No
8	Yellow	Coldplay	6	Yes
9	Late Night Talking	Harry Styles	7	Yes

[6] https://charts.spotify.com/charts/overview/global
[7] The author rated each song with a scale from 1 to 10. Songs that received a rating of 7 or higher were considered suitable for inclusion in the author's library.

Order	Track	Artist	Rating	Added
10	Without Me	Eminem	8	Yes (already in library)
11	Gato de Noche	Ñengo Flow, Bad Bunny	5	No
12	10:35	Tiësto, Tate McRae	8	Yes (already in library)
13	LOKERA	Rauw Alejandro, Lyanno, Brray	6	No
14	ANTIFRAGILE	LE SSERAFIM	5	No
15	Starboy	The Weeknd, Daft Punk	9	Yes (already in library)
16	Until I Found You	Stephen Sanchez	5	No
17	Neverita	Bad Bunny	4	No
18	DESPECHÁ	ROSALÍA	6	No
19	I'm Good (Blue)	David Guetta, Bebe Rexha	6	No
20	OMG	NewJeans	5	No

Table 4: Comparison of SVR and FNN Models on new test set

Training function	Number of songs added to library	Number of songs already to library	Total number of songs added to library
SVR	6	1	7
FNN	3	3	6

According to the tables presented, under SVR model, seven out of 20 recommended songs were added to the author's library, with one of them already being present in the library. Under FNN model, six out of 20 songs were added to the author's library, with three of them already being present in the library. If we define the accuracy rate as the total number of songs that were added to library divided by the total number of recommended songs, we could see that the SVR and FNN model had an accuracy rate of 35% and 30% respectively. These results suggested that both models performed well, given that the author typically only adds up to three songs from the list of 30 songs recommended from Spotify Discover Weekly. Moreover, the fact that two songs, *I'm Good (Blue)* and *LOKERA*, were recommended by both models, which added the credibility to two models.

4. Conclusion

The purpose of this paper is to compare the performance of SVR and FNN models for predicting streaming time of tracks based on their respective audio features. The performance of the developed models was evaluated via RMSE, nRMSE, R-squared and accuracy rate. Based on the summary of the table below, it was shown that both SVR and FNN models had

relatively small RMSE and nRMSE values, with the FNN model performing slightly better. Although the R-squared values for both models were not favorable, additional analysis by testing the models in new dataset showed that the SVR model had an accuracy rate of 35%, while the FNN model had an accuracy rate of 30%. These results suggested that both models performed well in predicting streaming time based on audio features.

Table 5: Song Recommendation Performance: SVR vs FNN

Training function	RMSE	nRMSE	R-squared	Accuracy Rate
SVR	5,513,095	4.0%	-0.17	35%
FNN	5,253,167	3.9%	-0.07%	30%

5. Future Research Directions

This study primarily employed a content-based filtering method for Spotify recommendation system, focusing on predicting song streaming time of the songs based on their audio features. The research provided valuable insights into the performance of SVR and FNN models and highlighted the importance of using multiple measures to evaluate model performance. The findings of this study may have implications for the development of more accurate models for streaming time prediction, which could captivate the attention of researchers and practitioners in the field of music technology.

Furthermore, potential future research directions may include addressing the cold start problem, particularly when implementing collaborative filtering method. The cold start problem refers to the challenges of generating recommendations for new users with limited data. Researchers could explore leveraging initial user preferences, demographic information or utilizing some external data sources to improve early-stage recommendations. Another research direction could involve investigating solutions to mitigate the limited diversity issue. Algorithmically driven listening through content-based recommendations is often associated with diminished consumption diversity. Exploring strategies such as serendipity-driven

recommendation, diversity-aware algorithms, multi-objective optimization, and periodic re-evaluation for recommending content that caters to users' short-term preferences while concurrently maintaining long-term diversity in users' consumption patterns could be a compelling area for further research.

Reference

Achieng, K. (2019). Modelling of soil moisture retention curve using machine learning techniques: artificial and deep neural networks vs support vector regression models. Computers & Geosciences, *133*. https://doi.org/10.1016/j.cageo.2019.104320

Bhattacharya, S., Kalita, K., Cep, R., & Chakraborty, S. (2021). A comparative analysis on prediction performance of regression models during machining of composite materials. *Materials (Basel)*, *14*(21), 6689. https://doi.org/10.3390/ma14216689

Chen, F., Li, H., Xu, Z., Hou, S. & Yang D. (2015). User-friendly optimization approach of fed-batch fermentation conditions for the production of iturin A using artificial neural networks and support vector machine. *Electronic Journal of Biotechnology*, *18*(4), 273-280. https://doi.org/10.1016/j.ejbt.2015.05.001

Kamehkhosh, I., Bonnin, G., & Jannach, D. (2019). Effects of recommendations on the playlist creation behavior of users. *User Modeling and User-Adapted Interaction*, *30*(3), 285-322. https://doi.org/10.1007/s11257-019-09237-4

Naina, Y., Anil, K. & Sukomal, P. (2022). Improved self-attentive musical instrument digital interface content-based music recommendation system. *Computational Intelligence*, *38*(4), 1232-1257. https://doi.org/10.1111/coin.12501

Rouf, N., Malik, M. B., Arif, T., Sharma, S., Singh S., Aich, S. & Kim, H. C. (2021). Stock market prediction using machine learning techniques: a decade survey on

methodologies, recent developments, and future directions. *Electronics*, *10*(21), 27-17. https://doi.org/10.3390/electronics10212717

Saufie, A. Z. U., Yahya, A. S., Ramli, N. A. & Hamid, H. A. (2011). Comparison between multiple linear regression and feedforward backpropagation neural network models for predicting PM_{10} concentration level based on gaseous and meteorological parameters. *International Journal of Applied Science and Technology*. *1*(4), 263-271. https://doi.org/10.1016/j.proenv.2014.03.033

Van den Oord, A., Dieleman, S., & Schrauwen, B. (2013). Deep content-based music recommendation. *Electronics and Information Systems department (ELIS), Ghent University.* https://papers.nips.cc/paper/5004-deep-content-based-music-recommendation.pdf