# ECE 4424/CS 4824 Final Project Report

**Group Participants:** Rahul Menon, Vivan Chopra, Umme Fatema Piu

**Topic:** Classification of Music Genre.

For our final project, our group chose to investigate the problem of how Music Genres could be classified utilizing Machine-Learning algorithms. The genre classification problem intrigued us as, nowadays, popular streaming services such as Spotify and Apple Music have massive data sets that include various genres. Furthermore, Spotify has a self-upload service, so there are algorithms to check if certain musical pieces fit the genre they have been uploaded into. The ideology behind this technology and how it affects our everyday life permitted us to explore it further.

Furthermore, we decided to investigate some Machine Learning Algorithms that could solve this issue. We then found out how the K-Nearest Neighbors Algorithm works to classify music and how the Support Vector Machine Algorithm is also utilized.

Moreover, we chose the data set and then decided that we needed to calculate the distance by finding the K number of neighbors from our previous research. As a group, we agreed that we needed to develop those functions in python utilizing the NumPy library that we previously used in class. We also decided to utilize the Support Vector Machine Algorithm to make comparisons on how each algorithm supports the issue at hand.

Finally, in order to evaluate the performance of the models we used accuracy, F-score. We tested the accuracy and precision of the model and then added an accuracy calculator function that tested the correct predictions by the total number of predictions. The section below shows our exact pipeline of what we followed to achieve our goal.

**Data collection/generation:**
We then looked for a dataset that we could use to classify the music of different genres. Finally, we needed to ensure that testing could be carried out efficiently. The data set that we finally selected was (the "GTZAN Dataset - Music Genre Classification") genre collection dataset as it had a large selection of genres, covering hip-hop, jazz, rock, and many others, which would create a good mix for classification.

**Data transformation/pre-processing:**
We have our labels as categorical variables so convert them to numerical variables. We had features in different ranges, so we have scaled our features to prevent dominance of one attribute over the others.

**Feature extraction/engineering:**
We had two csv files (30 seconds interval and 3 seconds interval) containing features of the audio files. For each song a mean and variance computed over multiple features that can be extracted from an audio file. We have used a 30 seconds interval length file for this project.

**Data split:**
We decided to separate our training and test data (70%-30%) respectively to ensure that test data is not biased and we achieve the most fair results possible.

**Model tuning/HPO:**
For KNN, we needed to calculate the optimum K value for the model. We have cross validated our training data to get optimum K value. We tried both 5 folds and 10 folds, in both cases the optimum K value is 4. Figure 1 shows accuracy for training and test dataset for different k-values. Though here K=5 seems to give more test accuracy, we still choose K= 4 as we have decided to prioritize our cross validation.

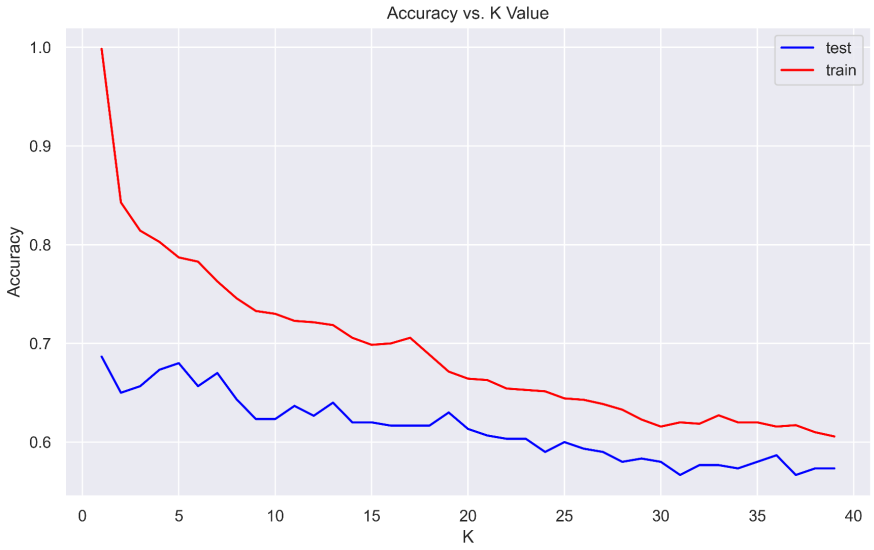


**Figure 1: Accuracy for training and test dataset vs K-value**

Our dataset is non linearly separable, so we have used a radial basis fitted (RBF) kernel for our SVM model. For SVM, we had a C parameter that trades off correct classification of training examples against maximization of the decision function's margin. And we had the Gamma parameter of RBF that controls the distance of influence of a single training point. Low values of gamma indicates a large similarity radius which results in more points being grouped together. For high values of gamma, the points need to be very close to each other in order to be considered in the same group (or class). Therefore, models with very large gamma values tend to overfit. For our model optimum C value is 10 and Gamma is 0.01. Figure 2 and Figure 3 shows accuracy for training and test dataset for different C and gamma parameters.

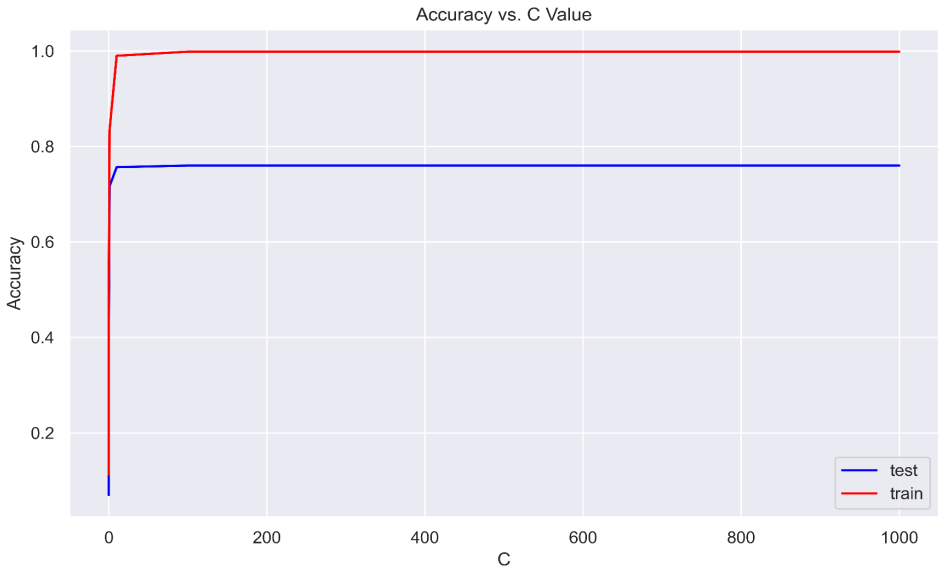


**Figure 2: Accuracy for training and test dataset vs C value in SVM**

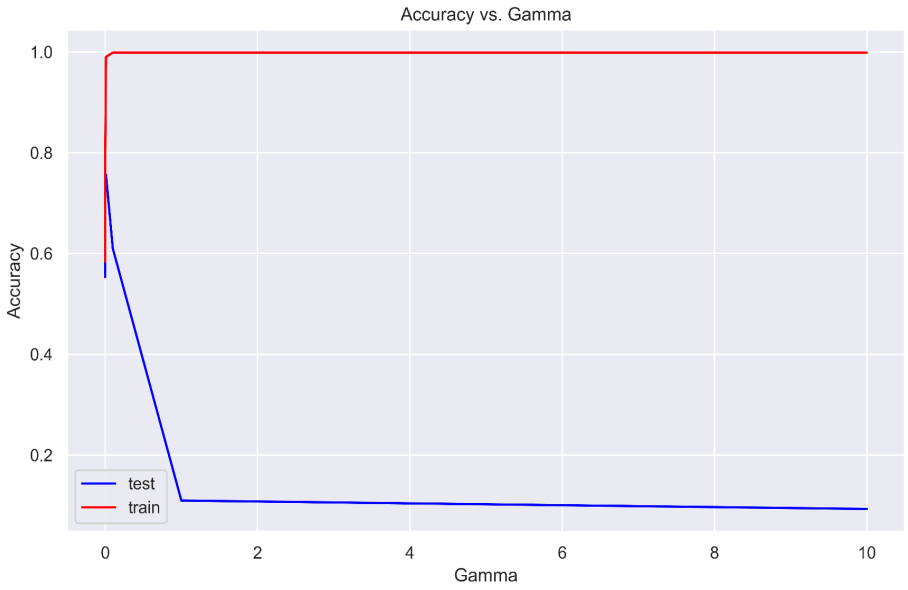


**Figure 3: Accuracy for training and test dataset vs Gamma in SVM**

**Performance evaluation (numbers and illustrating figures):**

Figure 4 shows performance evaluation for KNN and Figure 5 shows performance evaluation for the SVM. As seen by the results that are displayed by Figure 4 and Figure 5 it is possible to see that the Support Vector Machine algorithm produces a more accurate result as evaluated by the F-Score. So in terms of classification the SVM does a better job in classifying the data-set.

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.61 | 0.81 | 0.69 | 21 |
| 1 | 0.82 | 0.90 | 0.86 | 31 |
| 2 | 0.46 | 0.84 | 0.59 | 25 |
| 3 | 0.39 | 0.46 | 0.42 | 26 |
| 4 | 0.75 | 0.64 | 0.69 | 33 |
| 5 | 0.86 | 0.53 | 0.65 | 34 |
| 6 | 0.93 | 0.82 | 0.87 | 33 |
| 7 | 0.84 | 0.79 | 0.81 | 33 |
| 8 | 0.59 | 0.53 | 0.56 | 36 |
| 9 | 0.65 | 0.46 | 0.54 | 28 |
| accuracy | | | 0.67 | 300 |
| macro avg | 0.69 | 0.68 | 0.67 | 300 |
| weighted avg | 0.71 | 0.67 | 0.68 | 300 |

**Figure 4: Performance evaluation for KNN**

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.59 | 0.90 | 0.72 | 21 |
| 1 | 0.90 | 0.87 | 0.89 | 31 |
| 2 | 0.79 | 0.76 | 0.78 | 25 |
| 3 | 0.65 | 0.65 | 0.65 | 26 |
| 4 | 0.76 | 0.79 | 0.78 | 33 |
| 5 | 0.78 | 0.74 | 0.76 | 34 |
| 6 | 0.85 | 0.88 | 0.87 | 33 |
| 7 | 0.93 | 0.76 | 0.83 | 33 |
| 8 | 0.69 | 0.61 | 0.65 | 36 |
| 9 | 0.62 | 0.64 | 0.63 | 28 |
| accuracy | | | 0.76 | 300 |
| macro avg | 0.76 | 0.76 | 0.75 | 300 |
| weighted avg | 0.77 | 0.76 | 0.76 | 300 |

**Figure 5: Performance evaluation for SVM**

**Contribution of each team member:**

**Rahul Menon**
Contributed on project documentation, code testing and approach with other team members.

**Vivan Chopra**
Contributed on project documentation and code testing and evaluation.

**Umme Fatema Piu**
Contributed on code for data processing, model building and evaluation for KNN and SVM with the other members. Worked on project documentation.

## References

"GTZAN Dataset - Music Genre Classification." *Kaggle*, https://www.kaggle.com/datasets/andradaolteanu/gtzan-dataset-music-genre-classification?select=Data. Accessed 3 December 2022.