**B. Tech Program**

**Course: Machine Learning Lab**

**Course Code: DS2231**

**Music Playing Style classification**

by

**Maheeka Dutta**

(Reg. No: **219309031**)

B.Tech. 2nd Year, Section-A

Under the guidance

of

**Mr. Venkatesh Gauri Shankar**

**Assistant Professor (Sr. Scale)**

Department of Information and Technology,

School of Information Technology, Manipal University Jaipur, Jaipur

***Department of Information Technology,***

***School of IT,***

***Faculty of Engineering***

***Manipal University Jaipur, India***

**April 2023**

**Certificate**

This is to certify that the project titled **“Music Playing Styles Classification”** is a record of the bona fide work done by **Maheeka Dutta** (Reg No:219309031) submitted for the partial fulfilment of the requirements for the completion of the Machine Learning Lab (DS2231) course in the **Department of Information Technology** of **Manipal University Jaipur,** during the academic session March - June 2023.

|  |
| --- |
| *Signature of the mentor* |
| Venkatesh Gauri Shankar,  Assistant Professor, (Senior Scale)  Department of IT, SCIT |

**Introduction**

The aim of this project is to predict the music playing styels using machine learning algorithms on available historical data. Music is classified into acquired/closed based on data like chroma, rolloff,etc. We apply different algorithms on the dataset and compare their accuracy.

**Methodology**

**1) K-Nearest Neighbor:**

**K-Nearest Neighbors Algorithm. The k-nearest neighbors algorithm, also known as KNN or k-NN, is a non-parametric, supervised learning classifier, which uses proximity to make classifications or predictions about the grouping of an individual data point.**

**dataset = pd.read\_csv('dataset.csv')**

**X = dataset.iloc[:, [1, 2, 3]].values**

**y = dataset.iloc[:, -1].values**

**from sklearn.preprocessing import LabelEncoder**

**le = LabelEncoder()**

**X[:,0] = le.fit\_transform(X[:,0])**

**from sklearn.model\_selection import train\_test\_split**

**X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size = 0.20, random\_state = 0)**

**from sklearn.preprocessing import StandardScaler**

**sc = StandardScaler()**

**X\_train = sc.fit\_transform(X\_train)**

**X\_test = sc.transform(X\_test)**

**from sklearn.neighbors import KNeighborsClassifier**

**classifier = KNeighborsClassifier(n\_neighbors = 5, metric = 'minkowski', p = 2)**

**classifier.fit(X\_train, y\_train)**

**2) Naive Bayes:**

**The Naïve Bayes classifier is a supervised machine learning algorithm, which is used for classification tasks, like text classification. It is also part of a family of generative learning algorithms, meaning that it seeks to model the distribution of inputs of a given class or category.**

**from sklearn.naive\_bayes import GaussianNB**

**classifier = GaussianNB()**

**classifier.fit(X\_train, y\_train)**

**from sklearn.metrics import confusion\_matrix,accuracy\_score**

**cm = confusion\_matrix(y\_test, y\_pred)**

**ac = accuracy\_score(y\_test,y\_pred)**

**3)Support Vector Machine:**

**Support vector machines (SVMs) are a set of supervised learning methods used for classification, regression and outliers detection. The advantages of support vector machines are: Effective in high dimensional spaces.**

**plt.scatter(X[:, 0], X[:, 1], c=y, cmap=plt.cm.Paired)**

**plt.xlabel('file name')**

**plt.ylabel('label')**

**plt.xlim(xx.min(), xx.max())**

**plt.title('SVC with linear kernel')**

**plt.show()**

**4)Hierarchial Clustering:**

**Hierarchical clustering is a popular method for grouping objects. It creates groups so that objects within a group are similar to each other and different from objects in other groups. Clusters are visually represented in a hierarchical tree called a dendrogram.**

**import scipy.cluster.hierarchy as sch**

**dendro = sch.dendrogram(sch.linkage(X, method = 'ward'))**

**plt.title('Dendrogram')**

**plt.xlabel('Genres')**

**plt.ylabel('Euclidean distances')**

**plt.show()**

**plt.scatter(X[y\_hc == 0, 0], X[y\_hc == 0, 1], s = 100, c = 'red', label = 'Cluster 1')**

**plt.scatter(X[y\_hc == 1, 0], X[y\_hc == 1, 1], s = 100, c = 'blue', label = 'Cluster 2')**

**plt.scatter(X[y\_hc == 2, 0], X[y\_hc == 2, 1], s = 100, c = 'green', label = 'Cluster 3')**

**plt.scatter(X[y\_hc == 3, 0], X[y\_hc == 3, 1], s = 100, c = 'cyan', label = 'Cluster 4')**

**plt.scatter(X[y\_hc == 4, 0], X[y\_hc == 4, 1], s = 100, c = 'magenta', label = 'Cluster 5')**

**plt.scatter(X[y\_hc == 5, 0], X[y\_hc == 5, 1], s = 100, c = 'yellow', label = 'Cluster 6')**

**plt.title('Clusters of genress')**

**plt.xlabel('filename')**

**plt.ylabel('label')**

**plt.legend()**

**plt.show()**

**RESULTS**

**1)K Nearest Neighbor**



**2)Naive Bayes**



**3)Support Vector Machine**





**4)Hierarchial Clustering**







**CONCLUSION**

We observe that KNN, Decision Tree and SVM have average accuracy (greater than 0.37)compared to others. By feeding new test data to the model, we can predict the labels of music with good accuracy.

**References**

**>**<https://docs.python.org/3/tutorial/errors.html>