**ML Project**

# **Introduction:**

Machine learning is the field of study that gives computers the capability to learn without being explicitly programmed. It is a subset of Artificial Intelligence (AI). Basically, machine learning is giving enough data to the computer/model to create or find patterns in the data which it will use when additional data is provided to get the results. This ability to learn without writing codes or program makes the machine act like human. Nowadays, it is being used almost in every field including self-driving cars, robotics, language processing, vision processing, forecasting stock market trends etc.

Machine Learning with Python:

A machine learning project involves various steps as follows:

1. **Import the Data:** Often comes in the form of a csv file. A database with lots of data which can be exported and stored in a csv file for the purpose of using it in machine learning project.
2. **Clean the Data:** Removing duplicates, irrelevant or null values to get accurate results and to avoid model to learn bad patterns. All text-based data must be converted into numeric values.
3. **Split the Data into Training/Test Sets:** Creating two segments, one for training the model and other for testing the model, to make sure that the model is working accurately/rightly.
4. **Create a Model:** Selecting an algorithm to analyze the data. Choosing an algorithm depends on the kind of problem which is being solved and the available input data.
5. **Train the Model:** Feed the data for training which will further look for any patterns.
6. **Make Predictions:** Based on patterns, the model will make predictions.
7. **Evaluate and Improve:** The prediction may/may not be accurate. Therefore, need to evaluate the predictions and measure the accuracy, then make changes to the model accordingly or select a different algorithm to get better results.

# **Problem Definition:**

In this project, we will use machine learning and build a model using google colab. We will feed this model with some sample data based on the existing datasets. The model will learn the patterns in the data and will ask to make the predictions.

# **Problem Solution:**

1. Importing the data: The csv file is obtained from Kaggle website.

Graphical user interface, text, application

Description automatically generated

Table

Description automatically generated

1. Cleaning or preparing the data to remove any duplicates or null values. Since this data is clean, we will skip this step. However, we will split this data into two separate sets, input dataset and output dataset.
2. Splitting the dataset as mentioned in the previous step:

Table

Description automatically generated

Table

Description automatically generated

1. Create a model using machine learning algorithm. The suitable library which provides the algorithm to analyze our data is scikit-learn.

A picture containing table

Description automatically generated

1. Making predictions using our model:

Graphical user interface, text, application

Description automatically generated

1. Measuring the accuracy of the model:

Graphical user interface, text, application, email

Description automatically generated

Changing the test size to 80%:

Graphical user interface, text, application, email

Description automatically generated

If test size is changed, 80% for testing. The accuracy drops as we are using very little for training this model. The more data and the cleaner data will give better result.

1. Model Persistence:

Graphical user interface, text, application, email

Description automatically generated

1. Visualizing a Decision Tree:

Graphical user interface, text, application, email

Description automatically generated

Code for the digraph tree:

*digraph Tree {*

*node [shape=box, style="filled, rounded", color="black", fontname="helvetica"] ;*

*edge [fontname="helvetica"] ;*

*0 [label="age <= 30.5\ngini = 0.778\nsamples = 18\nvalue = [3, 6, 3, 3, 3]\nclass = Classical", fillcolor="#e5fad7"] ;*

*1 [label="age <= 25.5\ngini = 0.75\nsamples = 12\nvalue = [3, 0, 3, 3, 3]\nclass = Acoustic", fillcolor="#ffffff"] ;*

*0 -> 1 [labeldistance=2.5, labelangle=45, headlabel="True"] ;*

*2 [label="gender <= 0.5\ngini = 0.5\nsamples = 6\nvalue = [0, 0, 3, 3, 0]\nclass = Dance", fillcolor="#ffffff"] ;*

*1 -> 2 ;*

*3 [label="gini = 0.0\nsamples = 3\nvalue = [0, 0, 3, 0, 0]\nclass = Dance", fillcolor="#39e5c5"] ;*

*2 -> 3 ;*

*4 [label="gini = 0.0\nsamples = 3\nvalue = [0, 0, 0, 3, 0]\nclass = HipHop", fillcolor="#3c39e5"] ;*

*2 -> 4 ;*

*5 [label="gender <= 0.5\ngini = 0.5\nsamples = 6\nvalue = [3, 0, 0, 0, 3]\nclass = Acoustic", fillcolor="#ffffff"] ;*

*1 -> 5 ;*

*6 [label="gini = 0.0\nsamples = 3\nvalue = [3, 0, 0, 0, 0]\nclass = Acoustic", fillcolor="#e58139"] ;*

*5 -> 6 ;*

*7 [label="gini = 0.0\nsamples = 3\nvalue = [0, 0, 0, 0, 3]\nclass = Jazz", fillcolor="#e539c0"] ;*

*5 -> 7 ;*

*8 [label="gini = 0.0\nsamples = 6\nvalue = [0, 6, 0, 0, 0]\nclass = Classical", fillcolor="#7be539"] ;*

*0 -> 8 [labeldistance=2.5, labelangle=-45, headlabel="False"] ;*

*}*

The binary decision tree visual diagram is created using the graphviz.

Diagram

Description automatically generated

# **Conclusion:**

We have successfully created a model using machine learning which predicts the genre based on gender and age of the user with an accuracy of 75% to 100% which is pretty good considering the size of available data. The reason the accuracy fluctuates is that we are using little data for training this model, the more data we give to the model, better the accuracy. Lastly, a decision tree is created using scikit-learn library as shown in the problem solution. It creates a visual model of the predictions and how they work.