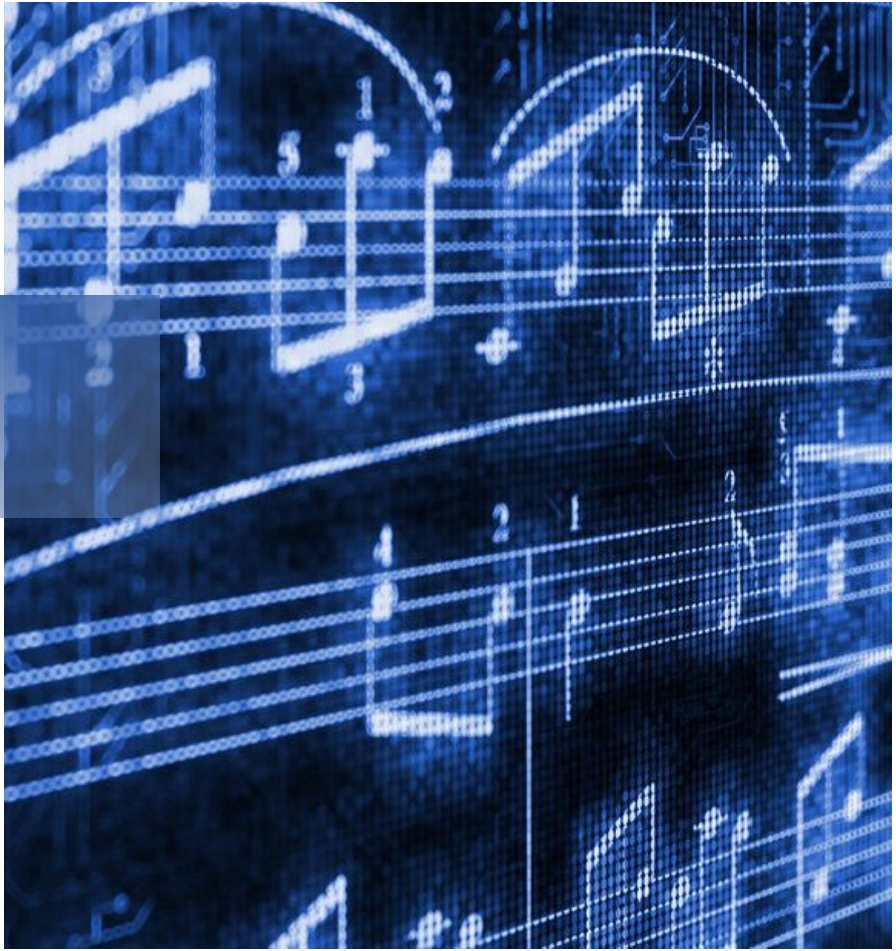
MUSIC GENRE CLASSIFICATION

Realized by : **EDDAMIR Amine - SOUIDI Yassine - CHAFIQUI Youssef**
Supervised by : **Mme. MOUSANNIF Hajar**

TABLE OF CONTENTS

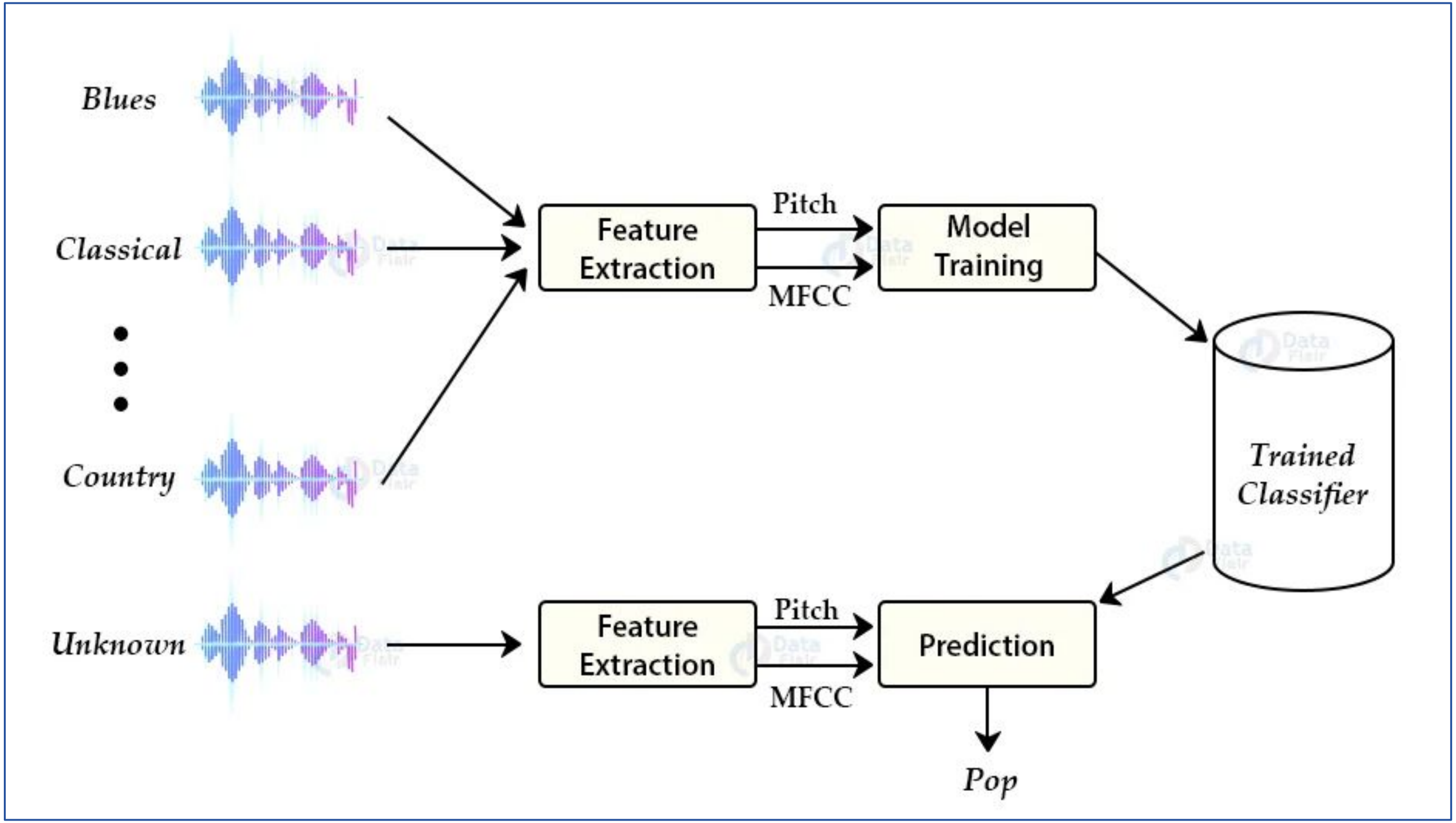


ABOUT THE PROJECT

Audio processing is one of the most complex tasks in data science as compared to image processing and other classification techniques. One such application is **Music Genre Classification** which aims to classify audio files in certain categories of music to which they belong.

The aim of the project is to avoid having to manually classify music into categories. To automate the process we use Machine Learning and Deep Learning algorithms and this is what we will implement in this project.

PROJECT PROCESS



GTZAN DATASET

GTZAN is a dataset which consists of audio tracks divided into 10 genres of music (Blues, Classical, Country, Disco, Hip-hop, Jazz, Metal, Pop, Reggae and Rock), each genre is represented by 100 tracks

In total, the dataset consists of **1000 audio files, each 30 seconds long.**

LIBROSA

Librosa is a python package for music and audio analysis. It provides the building blocks necessary to create music information retrieval systems.

We used librosa in our project to extract features from the music audio files.

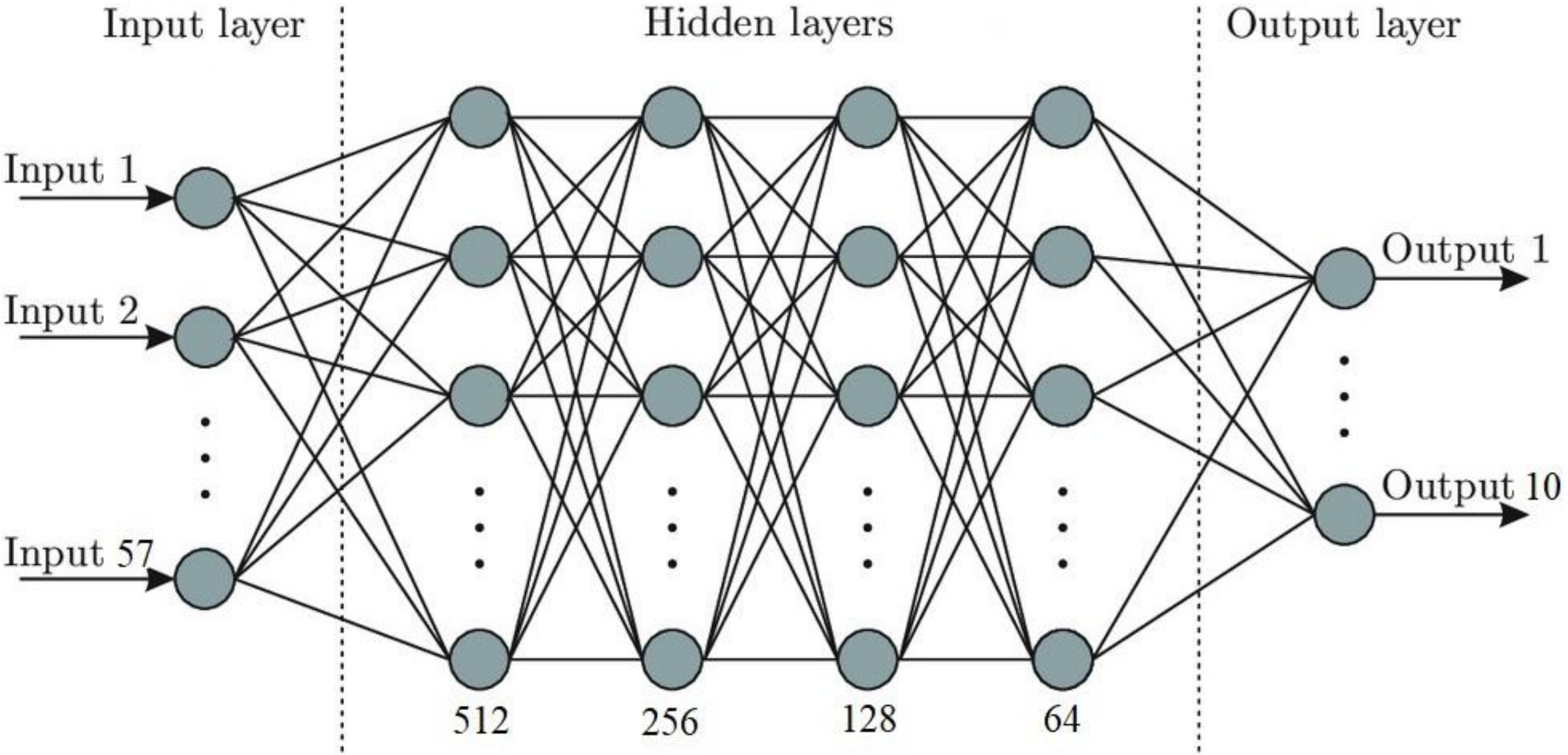
FEATURE EXTRACTION

1. Chromagram
2. Root-Mean-Square value (RMS)
3. Spectral Centroid
4. Spectral Bandwidth
5. Roll-off Frequency
6. Zero-crossing rate
7. Harmonic elements
8. Percussive elements
9. Tempo (Beats per minute)
10. Mel Frequency Cepstral Coefficient (MFCC)

MACHINE LEARNING MODELS

Model	Train Accuracy	Test Accuracy
Logistic Regression	90.1%	74.5%
Random Forest	100%	72.5%
XGBoost	100%	72%
CatBoost	100%	77.5%
Support Vector Machine	89.4%	76.5%
K-Nearest Neighbors	78.9%	71%

DEEP NEURAL NETWORK ARCHITECTURE



DEEP NEURAL NET MODEL CODE

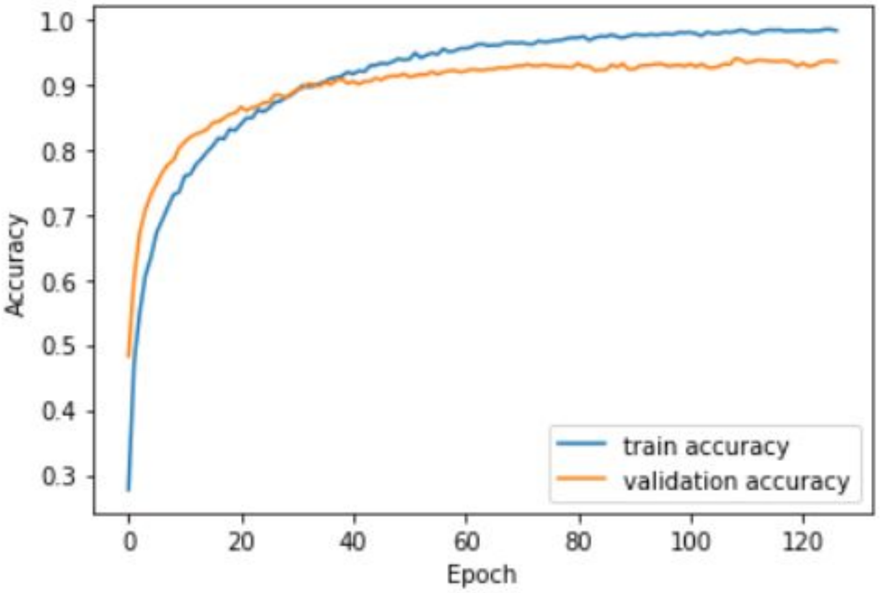
```
model = tf.keras.models.Sequential([
    tf.keras.layers.Dense(512, activation='relu', input_shape=[X_train.shape[1]]),
    tf.keras.layers.Dropout(0.2),
    tf.keras.layers.BatchNormalization(),

    tf.keras.layers.Dense(256, activation='relu'),
    keras.layers.Dropout(0.2),
    tf.keras.layers.BatchNormalization(),

    tf.keras.layers.Dense(128, activation='relu'),
    tf.keras.layers.Dropout(0.2),
    tf.keras.layers.BatchNormalization(),

    tf.keras.layers.Dense(64, activation='relu'),
    tf.keras.layers.Dropout(0.2),
    tf.keras.layers.BatchNormalization(),

    tf.keras.layers.Dense(10, activation='softmax'),
])
```

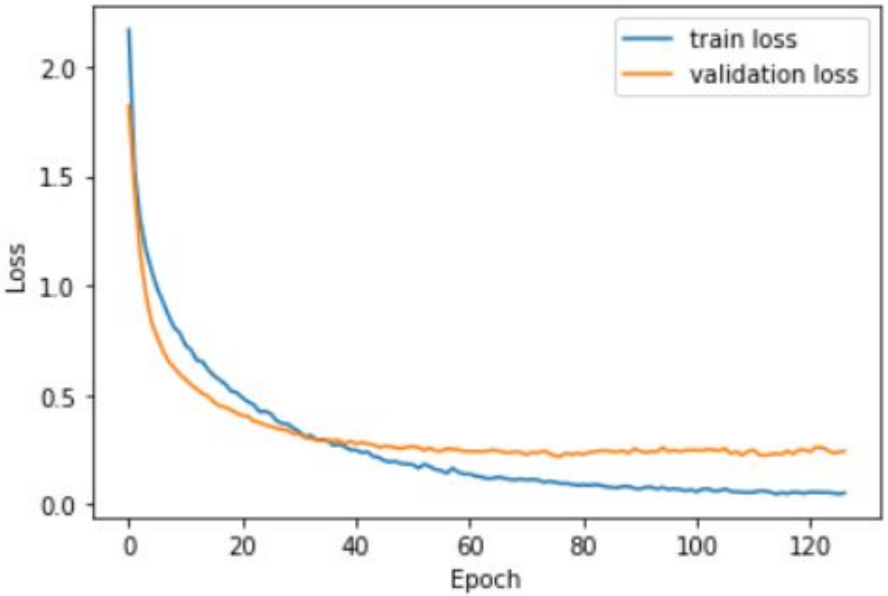
DEEP NEURAL NET MODEL CODE

```
opt = keras.optimizers.Adam(learning_rate=0.0003)
model.compile(
    optimizer=opt,
    loss='sparse_categorical_crossentropy',
    metrics='accuracy'
)
```

```
from tensorflow.keras import callbacks
early_stopping = callbacks.EarlyStopping(
    min_delta=0.001,
    patience=50,
    restore_best_weights=True
)
```

LEARNING CURVE

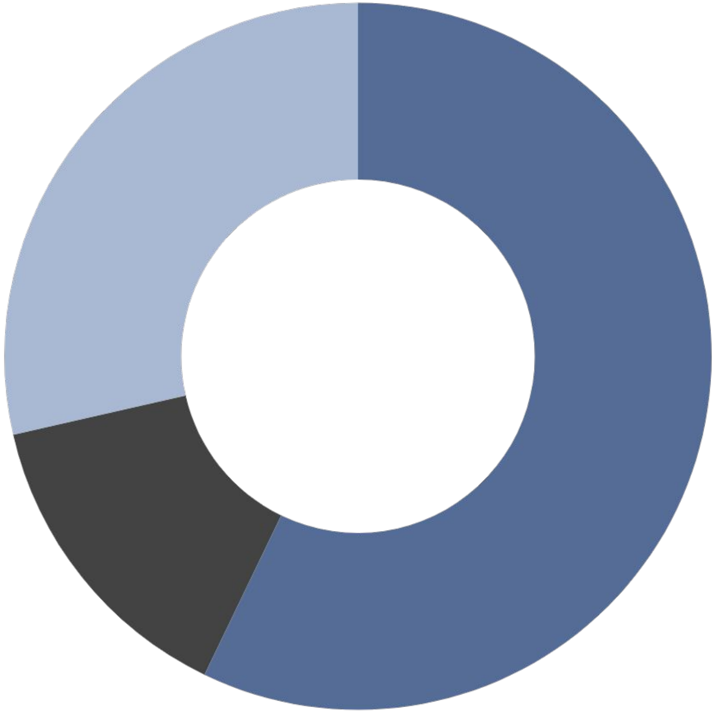




MODELS DEPLOYMENT

plotly | Dash

HEROKU



DATA PRE-PROCESSING & FEATURE EXTRACTION **60%**

MODELING **30%**

DEPLOYMENT **10%**

DEMONSTRATION

THANK YOU FOR YOUR ATTENTION