Project Report

On

Music Genre Detection using Machine Learning­

Submitted in partial fulfilment of the requirement for the

4th Semester

B. Tech Semester IV



Submitted To: Submitted By:

Dr. Narayan Chaturvedi Name: Parth Sarthi

Dept. of Computer Science University Roll No: 2017520

Section: CST SPL 1

Semester: 4th



Certificate

This is to certify that the project report entitled “Music Genre Detection using Machine Learning­” is a bona fide project work carried out by Parth Sarthi, University Roll Number 2017520. in partial fulfillment of the requirements for the IV Semester CSE is a record of my own work, under the supervision of Dr. Narayan Chaturvedi Sir, Assistant Professor, Department of Computer Science and Engineering of the Graphic Era University, Dehradun (Uttarakhand).

It is certified that all corrections/suggestions indicated for internal assessment have been incorporated. The project has been approved as it satisfies the academic requirements associated with the degree mentioned.

Parth Sarthi

2017520

Acknowledgement

I am submitting a project report on "Music Genre Detection using Machine Learning­" as per the scheme of Graphic Era Deemed University, Dehradun.

In this regard, I'd like to offer my heartfelt gratitude to our beloved Graphic Era University, as well as my genuine gratitude and debt of gratitude to Hon'ble Dr. Kamal Ghanshala, Chairman & President, Graphic Era University, Dehradun.

I'd want to offer my heartfelt gratitude to Dr. Devesh Pratap Singh, Head of the Department of Computer Science and Engineering, for creating a welcoming environment in which we could work and complete our research.

I want to express my gratitude for Dr. Narayan Chaturvedi Sir, Assistant Professor, Department of Computer Science and Engineering for her never-ending support and invaluable guidance at every stage and in every way which made this project possible.

Finally, I would like to express my gratitude to all of the faculty members of the Department of Computer Science and Technology, as well as my friends and parents, for their ongoing encouragement, support, and assistance throughout the the period of project conduction.

Parth Sarthi

CST SPL 1

2017520

Introduction

* **Introduction to Music Genre Classification**:

Audio classification is an application of Machine Learning where different sound is categorized in certain categories. Almost all data science enthusiasts want attractive and eye-catching data science projects on their resume and Audio processing is one such topic. In this project, we will build a complete music Genre classification project from scratch using a machine learning algorithm known as the K-Nearest Neighbours classification algorithm.

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 the audio files in certain categories of sound to which they belong. The application is very important and requires automation to reduce the manual error and time because if we have to classify the music manually then one has to listen out each file for the complete duration. So, to automate the process we use Machine learning and deep learning algorithms.

**Index**

|  |  |  |
| --- | --- | --- |
| Sl. No. | Title | Page Number |
| 1. | **Project Overview and Approach** | 6 |
| 2. | **Dataset Overview** | 6 |
| 3. | **Project Requirements** | 7 |
| 4. | **Methodology Followed** |  |
|  | * Libraries Installation | 8 |
|  | * Modules | 9 |
| 5. | **Implementation of Music Genre Classification Project** | 10 |
| 6. | **Conclusion** | 15 |
| 7. | **Reference** | 15 |

* **Project Overview and Approach**:

I can define my project problem statement as like given multiple audio files, and the task is to categorize each audio file in a certain category like audio belongs to Disco, hip-hop, etc. The music genre classification can be built using different approaches in which the top 4 approaches that are mostly used are listed below.

1. Multiclass support vector machine
2. K-Nearest Neighbours
3. K-means clustering algorithm
4. Convolutional neural network

K-Nearest Neighbour:

KNN is a machine learning algorithm used for regression, and classification. It is also known as the lazy learner algorithm. It simply uses a distance-based method to find the K number of similar neighbours to new data and the class in which the majority of neighbours lies, it results in that class as an output.

* **Dataset Overview**:

The dataset we will use is named the GTZAN genre collection dataset which is a very popular audio collection dataset. It contains approximately 1000 audio files that belong to 10 different classes. Each audio file is in .wav format (extension). The classes to which audio files belong are Blues, Hip-hop, classical, pop, Disco, Country, Metal, Jazz, Reggae, and Rock. You can easily find the dataset over Kaggle.

Project Requirements

Hardware Requirements:

* A Desktop/Laptop with good processor and Python/Jupyter installed.

Software Requirements:

* Python/Jupyter
* GTZAN
* Test Audios
* Visual Studio Code (IDE)

Methodology Followed

**Libraries Installation**:

1. NumPy:

NumPy (Numerical Python) is a Python library used for working with arrays. It also has functions for working in domain of linear algebra, Fourier transform, and matrices. It is an open-source project and you can use it freely.

1. Pandas:

Pandas is a Python library used for working with data sets. It has functions for analysing, cleaning, exploring, and manipulating data. The name "Pandas" has a reference to both "Panel Data", and "Python Data Analysis".

1. SciPy:

SciPy (Scientific Python) is a scientific computation library that uses NumPy underneath. It provides more utility functions for optimization, stats and signal processing. Like NumPy, SciPy is open source so we can use it freely.

1. python\_speech\_features:

This library provides common speech features for ASR including MFCCs and filterbank energies.

**Modules:**

* tempfile:

This module creates temporary files and directories. It works on all supported platforms.

* os:

This module provides a portable way of using operating system dependent functionality.

* math:

Python has a built-in module that you can use for mathematical tasks. The math module has a set of methods and constants.

* pickle:

Python pickle module is used for serializing and de-serializing python object structures.

* random:

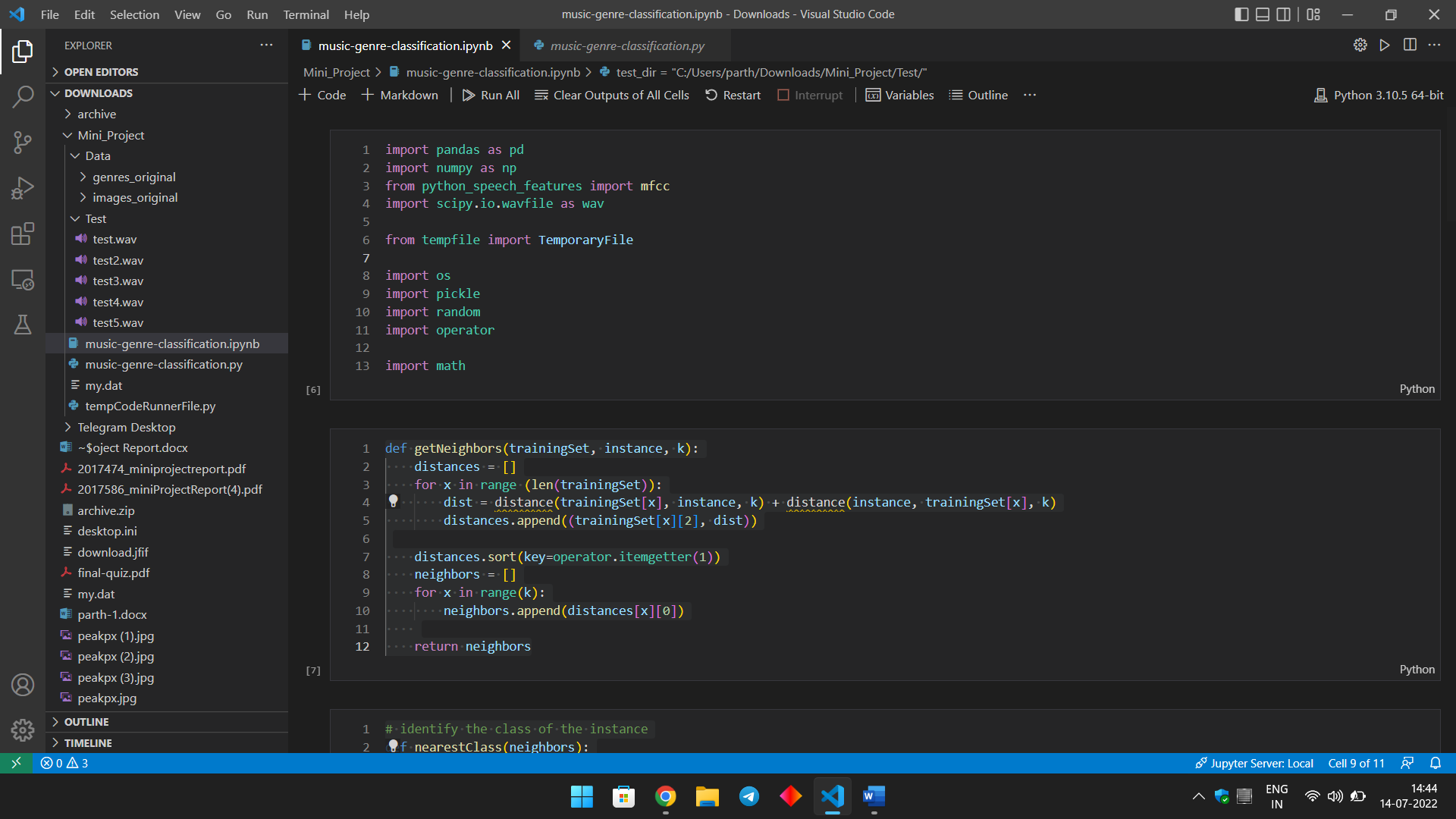
Python Random module is an in-built module of Python which is used to generate random numbers. These are pseudo-random numbers means these are not truly random.

* operator:

The Python operator module is one of the inbuilt modules in Python, and it provides us with a lot of functions which we can use to perform various mathematical, relational, logical and bitwise operations on two input numbers.

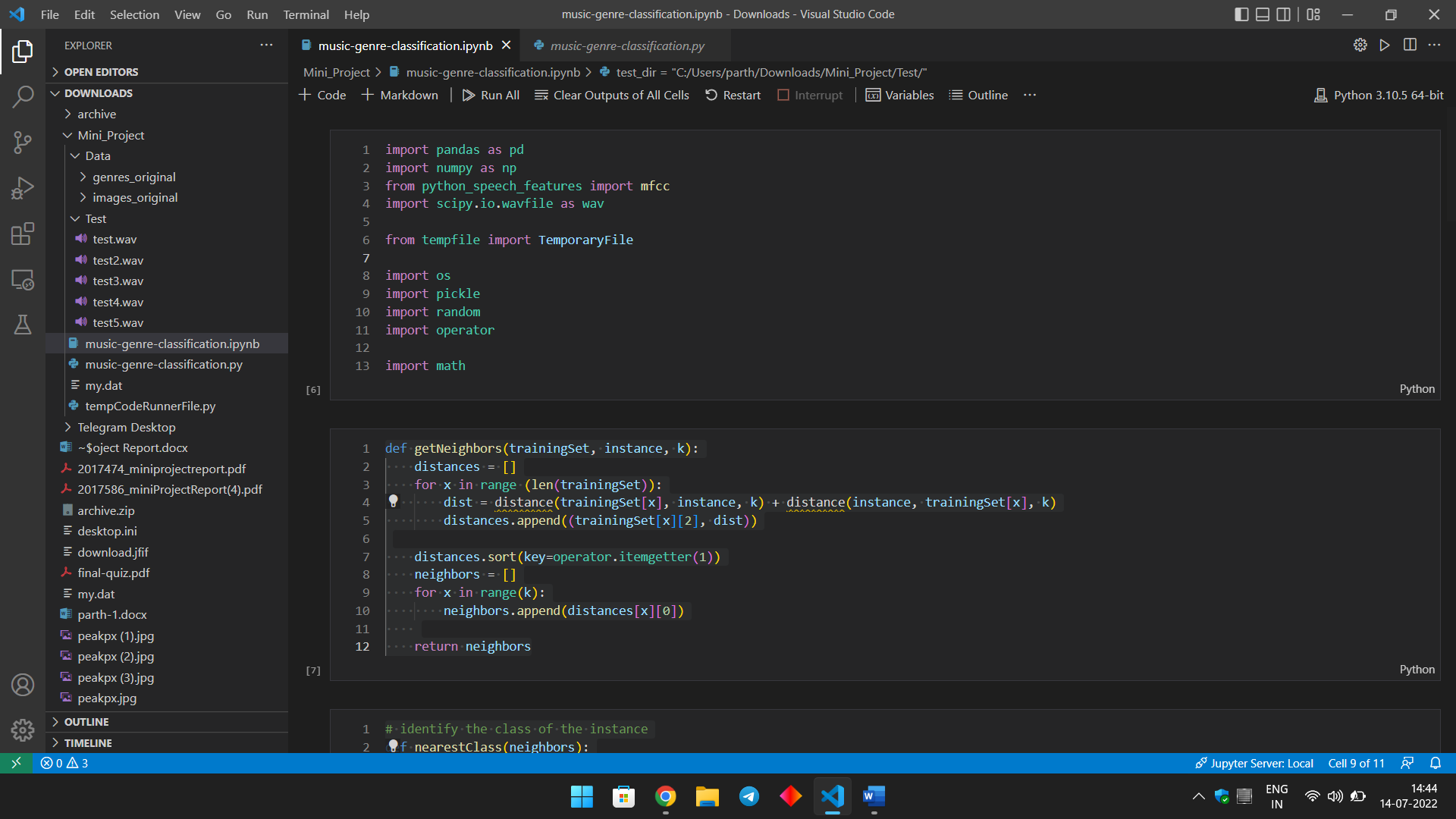
Implementation of Music Genre Classification Project

1. **Import Required Libraries:**



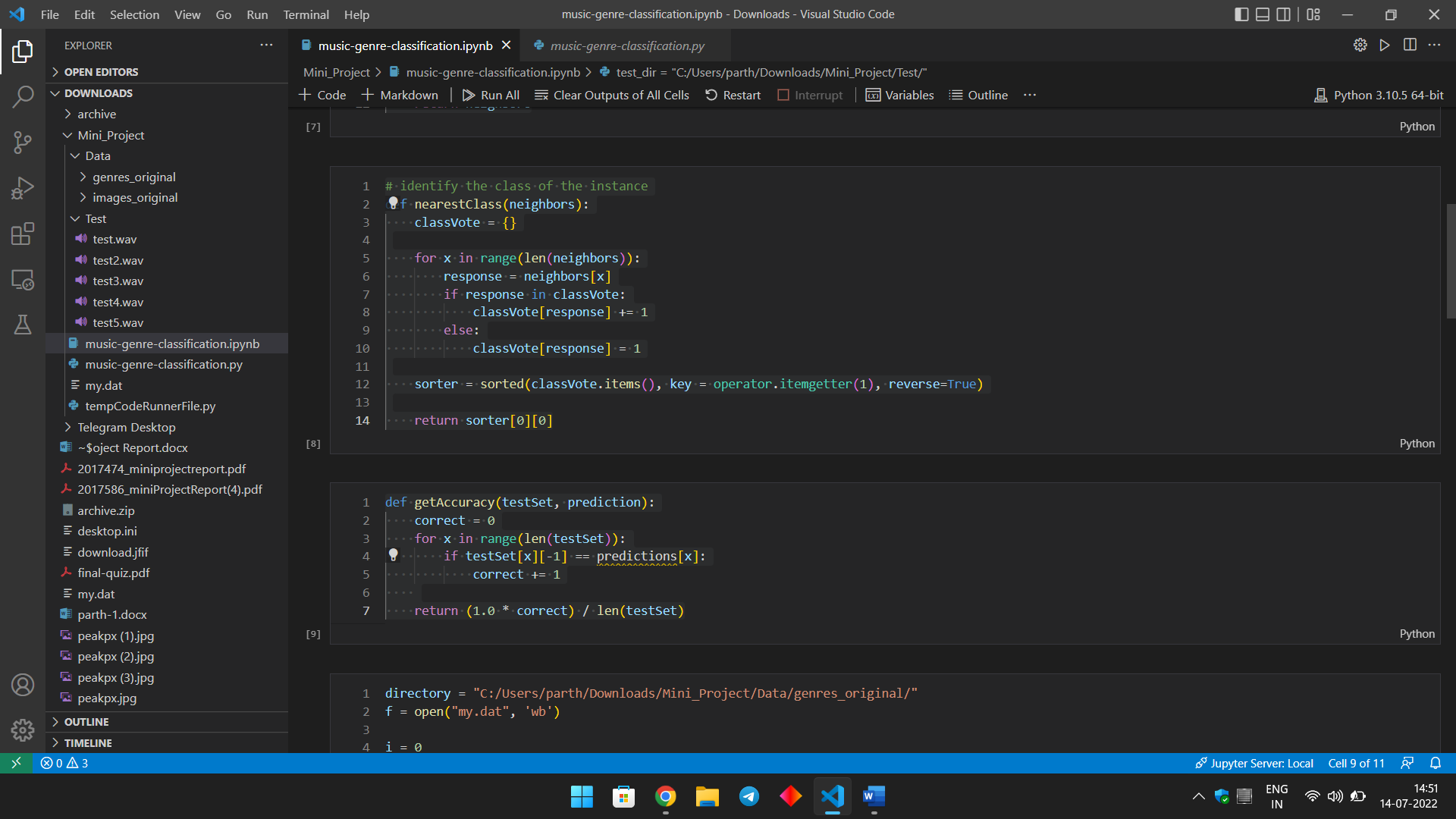
1. **Calculate distance between feature vectors, and find neighbours:**

It will find the distance of each point with every other point in training data and then we find all the nearest K neighbours and return all neighbours



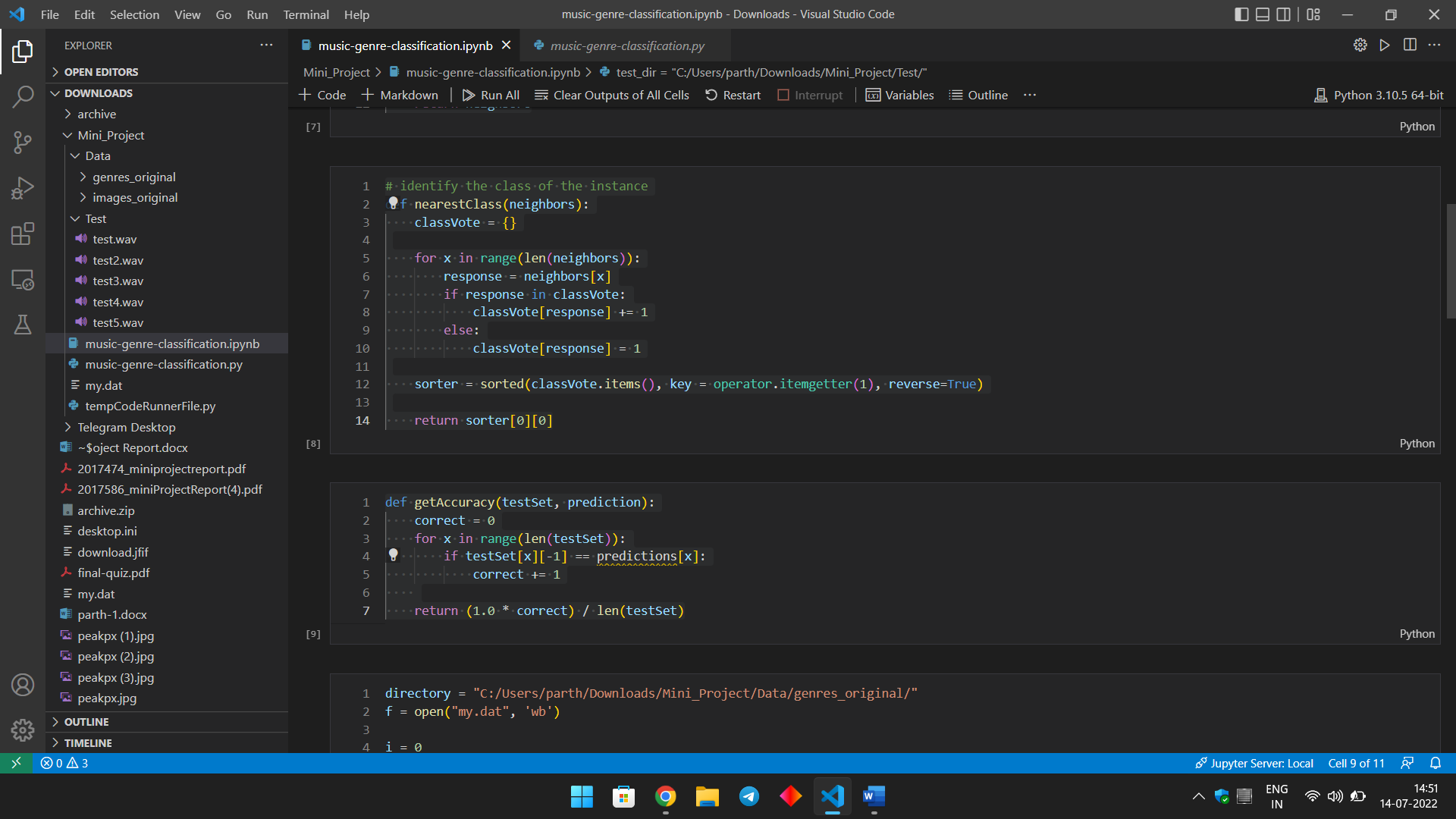
1. **Identify the class of nearest neighbours**:

After creating the frequency map we sort the map in descending order based on neighbours count and return the first class.



1. **Model Evaluation:**

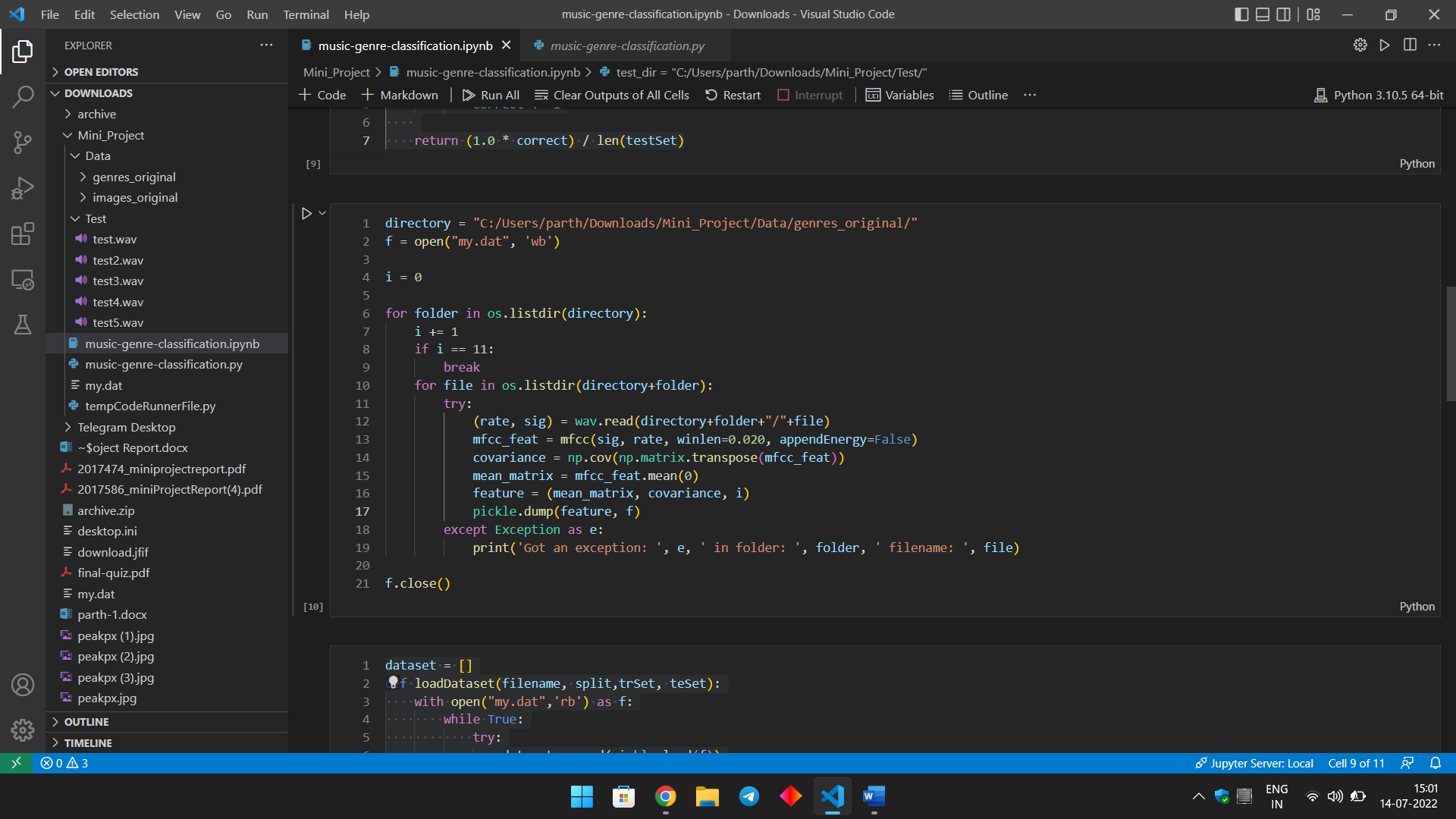
We also require a function that evaluates a model to check the accuracy and performance of the algorithm we build.



1. **Feature Extraction:**

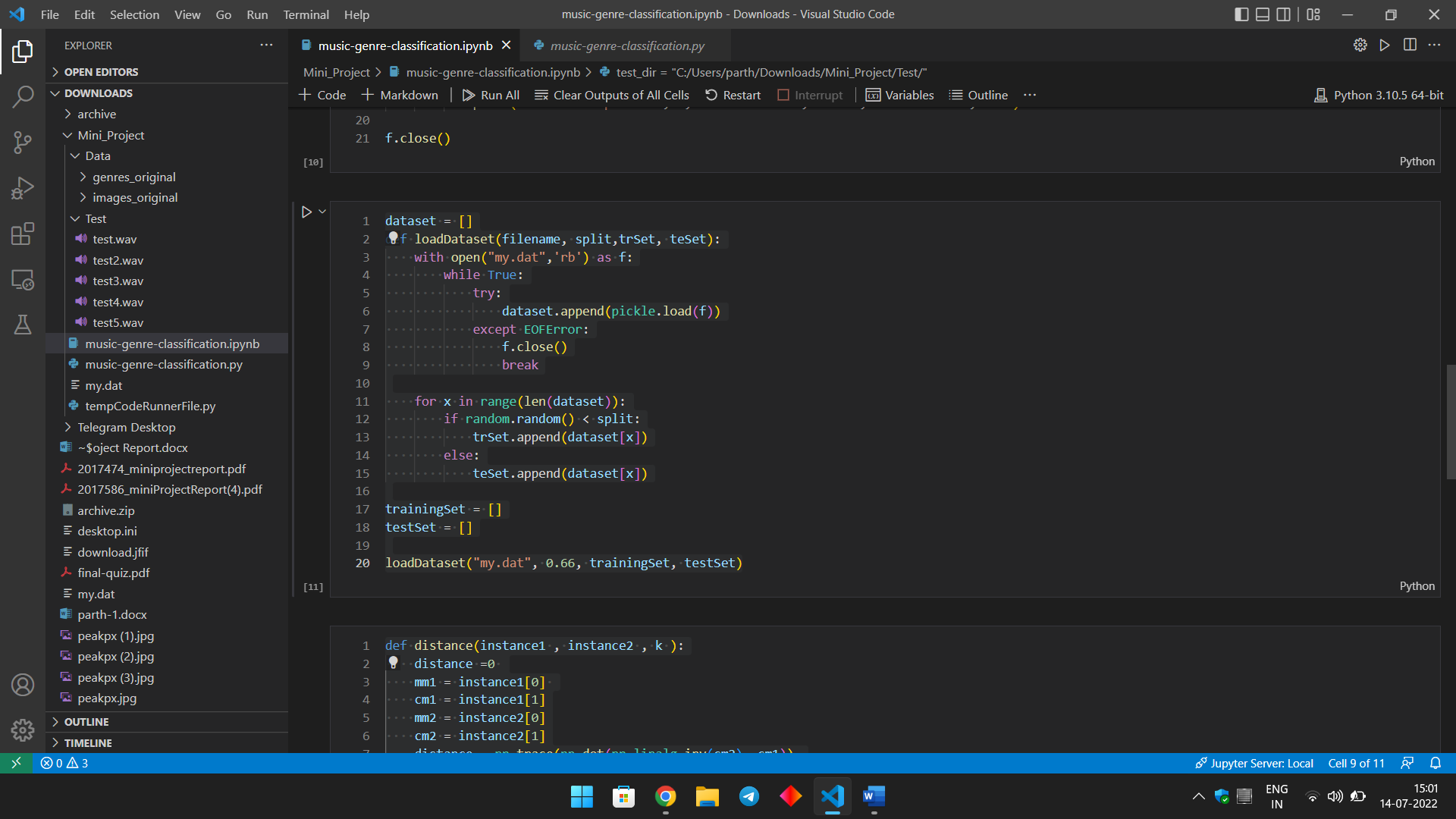
Feature extraction is a process to extract important features from data. It includes identifying linguistic data and avoiding any kind of noise. Audio features are classified into 3 categories high-level, mid-level, and low-level audio features.

We use MFCC (**Mel Frequency Cepstral Coefficients**) feature that helps in extract mid-level and low-level audio features as shown in below figure:



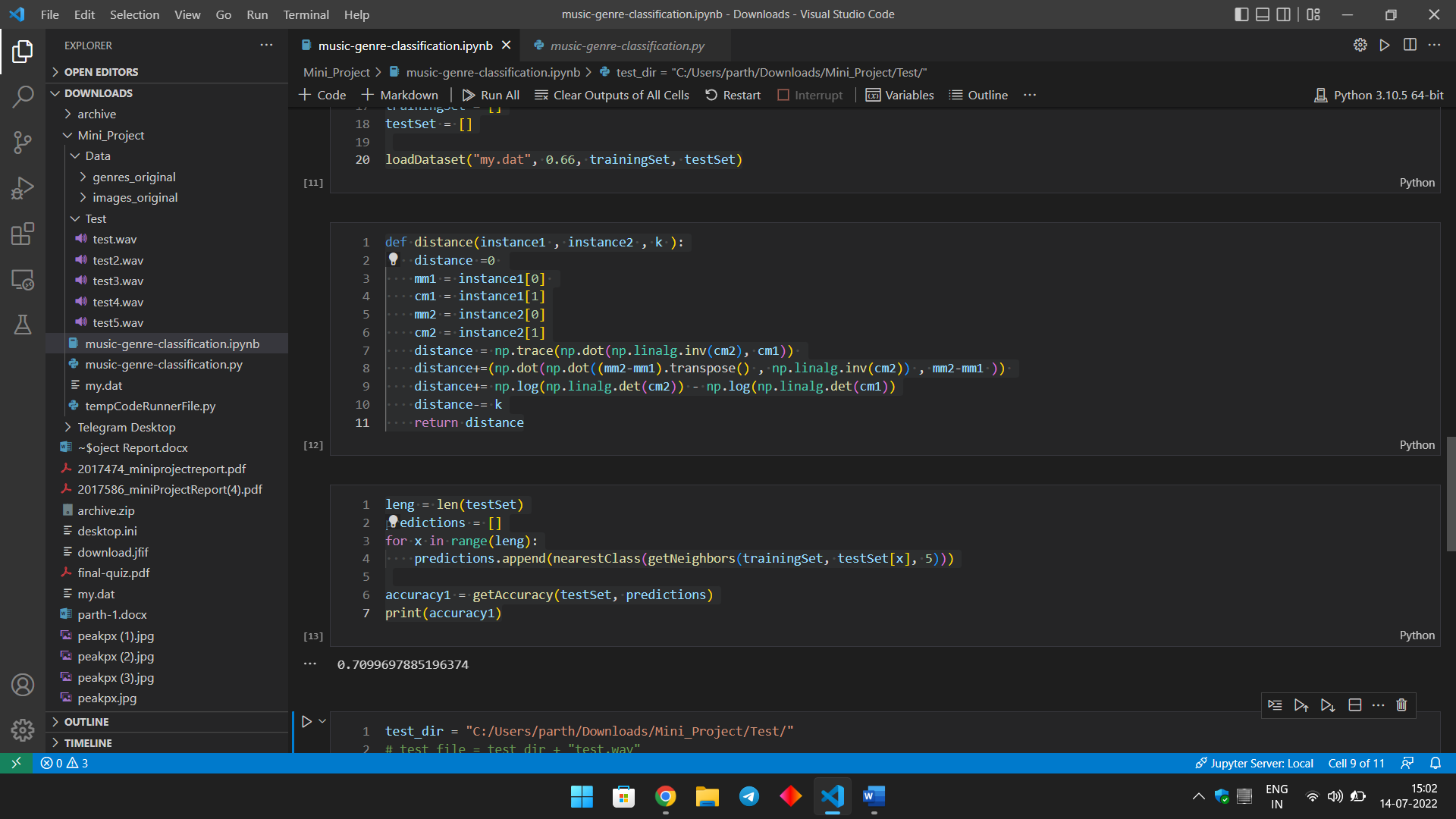
1. **Train-test split the dataset**:

There are different approaches to do train test split. here I am using a random module and running a loop till the length of a dataset and generate a random fractional number between 0-1 and if it is less than 66 then a particular row is appended in the train test else in the test set.



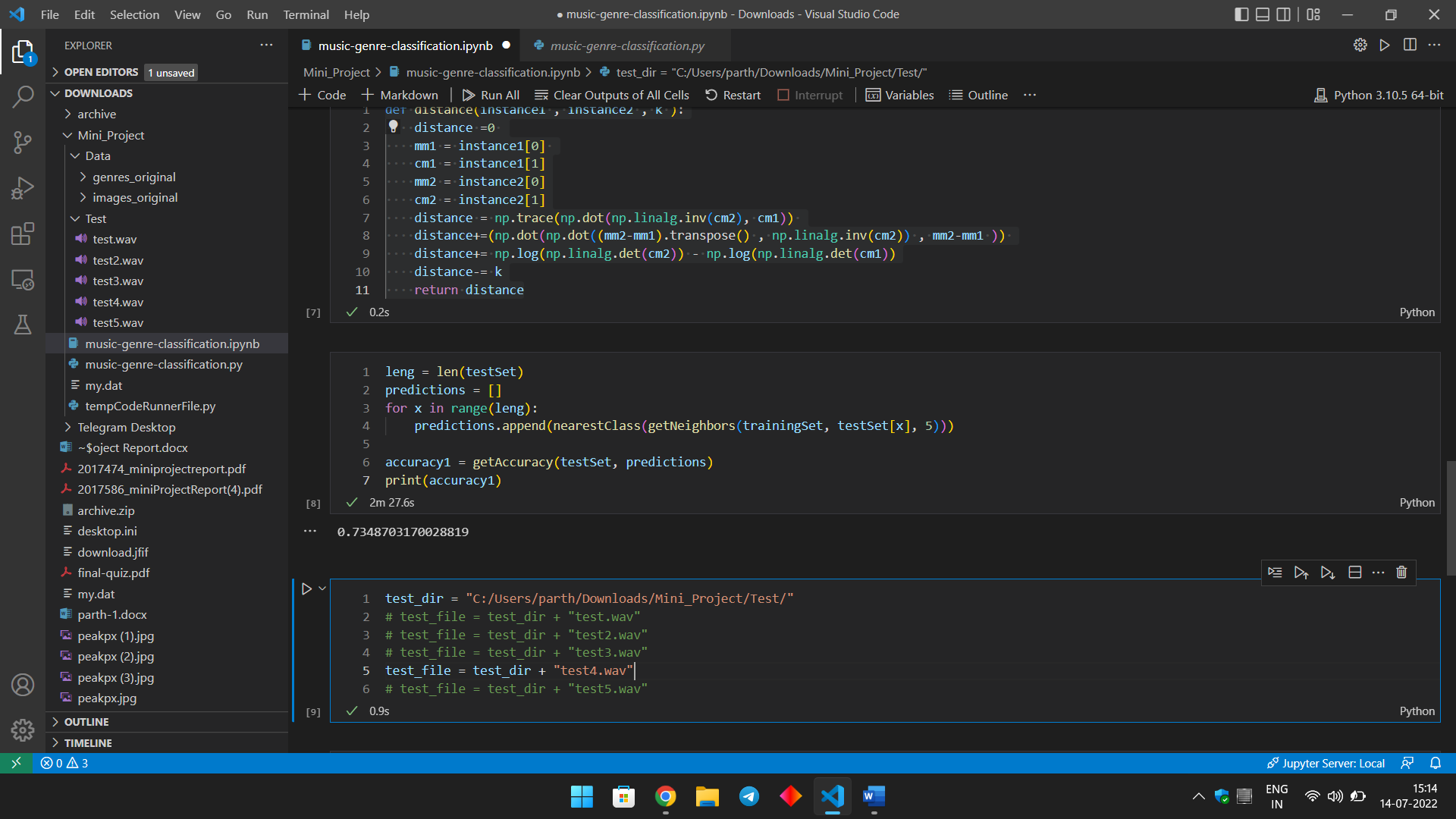
1. **Calculate the distance between two instance:**

As shown in the function accepts two data points (X, and Y coordinates) to calculate the actual distance between them. we use the numpy linear algebra package which provides a low-level implementation of standard linear algebra. So we first find the dot product between the X-X and Y-Y coordinate of both points to know the actual distance after that we extract the determinant of the resultant array of both points and get the distance.



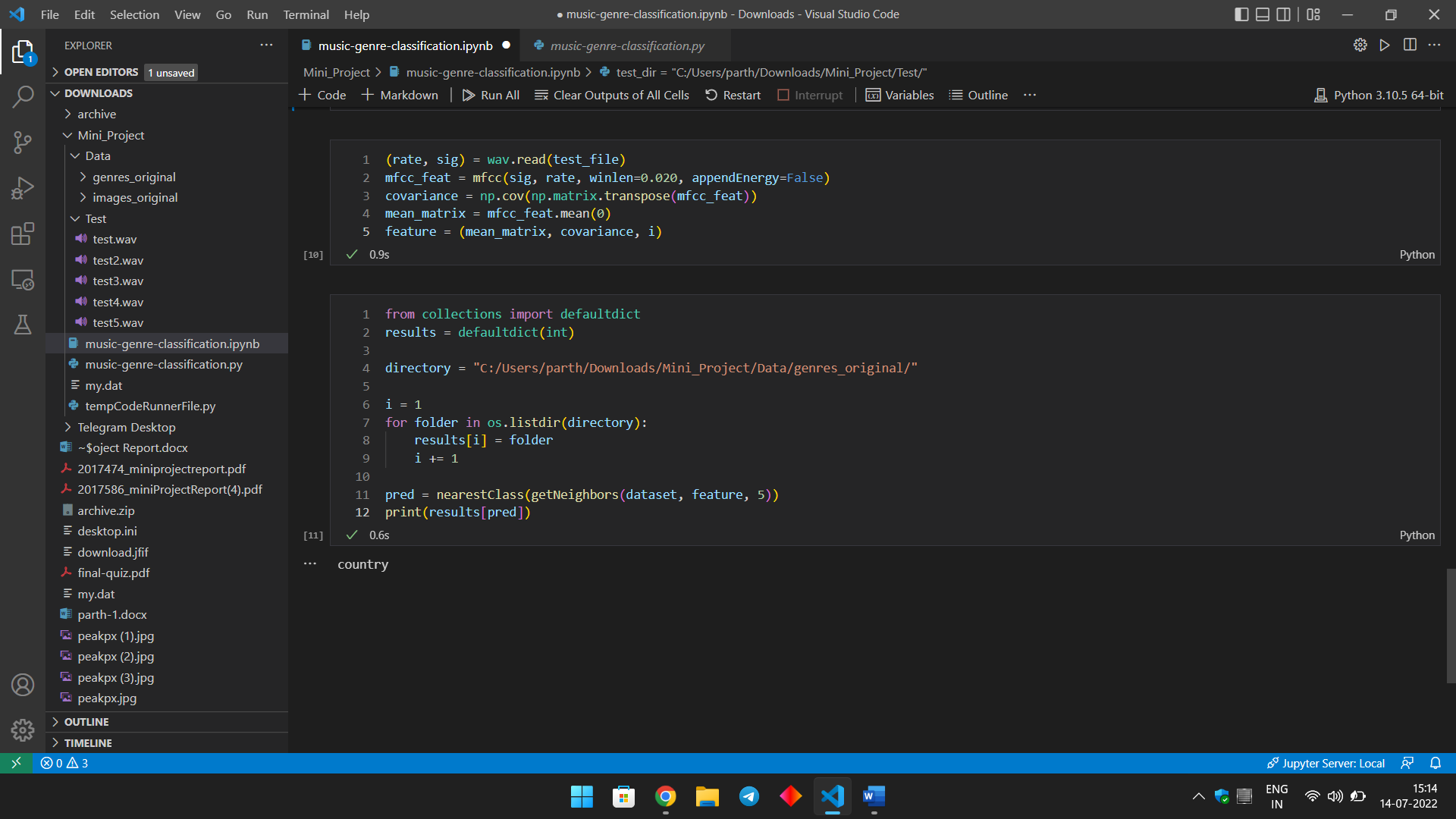
1. **Training the Model and making predictions**:

In this Step we feed the data to KNN algorithms and make all predictions and receive accuracy on the test dataset. We got the accuracy here is 0.7348703170028819.

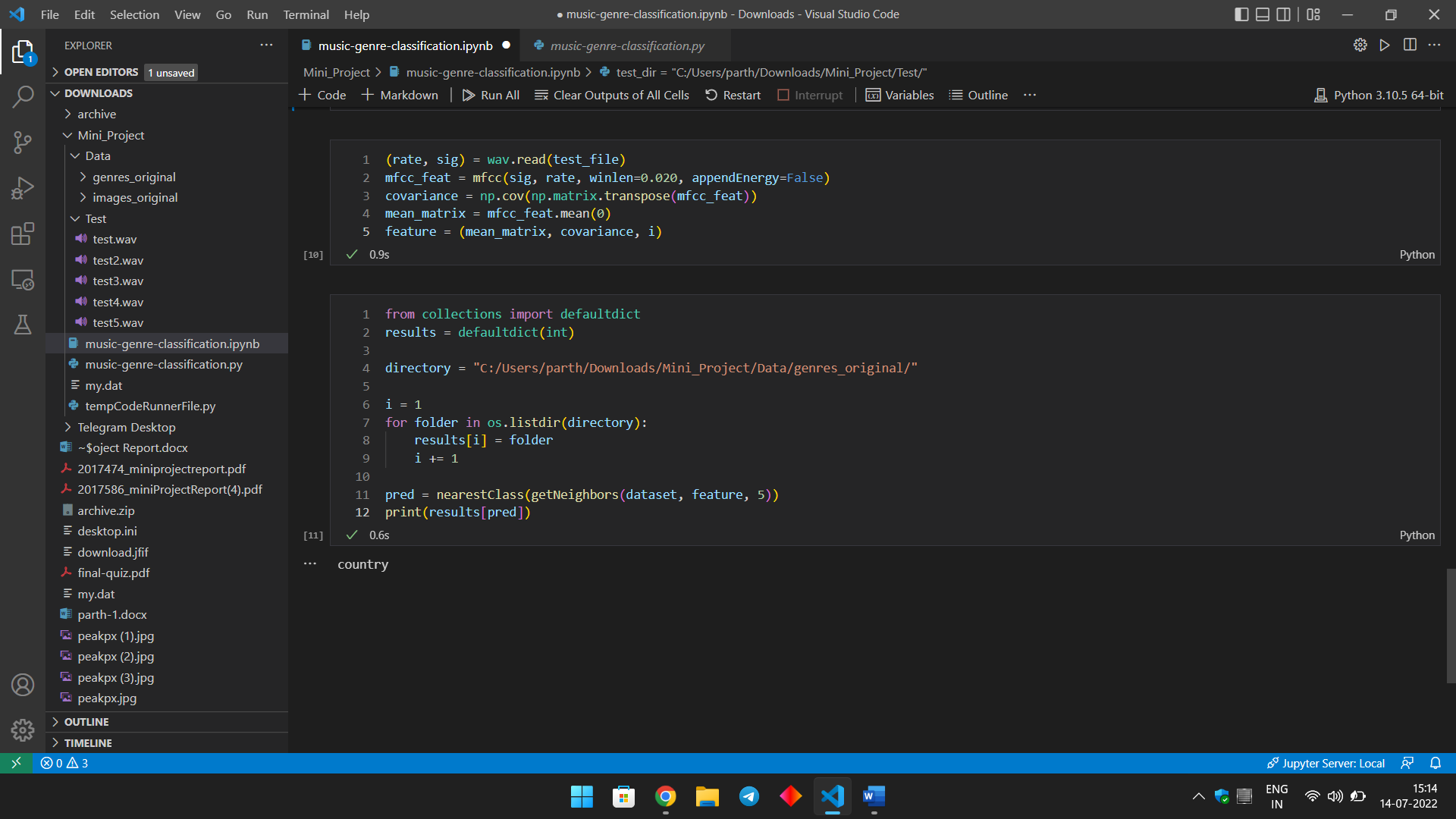


1. **Test the Classifier with the new Audio File**:

Here, we provide the Sample Video to the classifier and get output from Which category that audio is belong it may from jazz, classical or from possible Genres given in the data set as you see in below snapshot it predict the output on one sample audio.



Output of The Classifier on the Given Sample Audio:



Conclusion

We have started the project with the initial setup and used MFCC to extract features from audio files. After that, we have built a KNN classifier from scratch that finds K number of nearest neighbour based on features and maximum neighbour belonging to particular class gives as an output. We got approximately 73.4 percent accuracy on the model. After train the model and got accuracy, we use some sample audio files to test the classifier and predict the genre of that audio sample and got output correctly without any problem as you see in the snapshots.

Reference

1. <https://www.delftstack.com>
2. <https://www.etutorialspoint.com>
3. <https://www.geeksforgeeks.org>
4. <https://www.javatpoint.com>
5. <https://www.youtube.com>