

# Configuration Manual

MSc Research Project  
MSc Cloud Computing

Pradeep Manoharan Susila  
Student ID: X21121141

School of Computing  
National College of Ireland

Supervisor: Vikas Sahni

**National College of Ireland**  
**MSc Project Submission Sheet**



**School of Computing**

**Student Name:** Pradeep Manoharan Susila  
**Student ID:** 21121141  
**Programme:** Msc Cloud Computing **Year:** 2022-2023

**Module:** Msc Research Project

**Supervisor:** Vikas Sahni

**Submission Due Date:** 15/12/2022

**Project Title:** Effective Use of ML ops in Music Recommendation System

**Word Count:** 1079 Words **Page Count:** 8

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

**Signature: Pradeep Manoharan Susila**

**Date:** 15/12/2022

**PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST**

Attach a completed copy of this sheet to each project (including multiple copies)	<input type="checkbox"/>
<b>Attach a Moodle submission receipt of the online project submission,</b> to each project (including multiple copies).	<input type="checkbox"/>
<b>You must ensure that you retain a HARD COPY of the project,</b> both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	<input type="checkbox"/>

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

<b>Office Use Only</b>	
Signature:	
Date:	
Penalty Applied (if applicable):	

# Configuration Manual

Pradeep Manoharan Susila  
Student ID: 21121141

## 1 Music recommendation System using ML ops:

1. Implemented Machine Learning to train the datasets for recommending the music to the user. To automating the procedure, we have integrated the music recommendation system with the cloud. We are use the AWS sage maker, s3 bucket, and lambda in Amazon Web Services (AWS). By putting in place the sage maker, we are uploading all the CSV files and the ipynb file in the notebook. After that, all the data will be saved in the s3 bucket.
2. By putting in place the lambda function, whenever new data enters the system, the lambda will automatically trigger, and we will be able to get all the logs. With the cloud watch, it will be possible to record what kind of data has been uploaded, as well as when it has been uploaded, and access all the associated information.
3. If the newly supplied data from Spotify signifies that it will automatically train the data, then it will deploy in the system. We are using the information that was provided by Spotify, which comprises the listening habits of various people. We are going to take that data, and then we are going to clean the data. During that cleaning procedure, any data that is deemed undesirable will be deleted. Afterwards, we will get rid of the data that has a null value. We can circumvent the issue with the system's cold start by putting this into action.

The library packages imported initially to get use of it. The utilised packages are listed below:

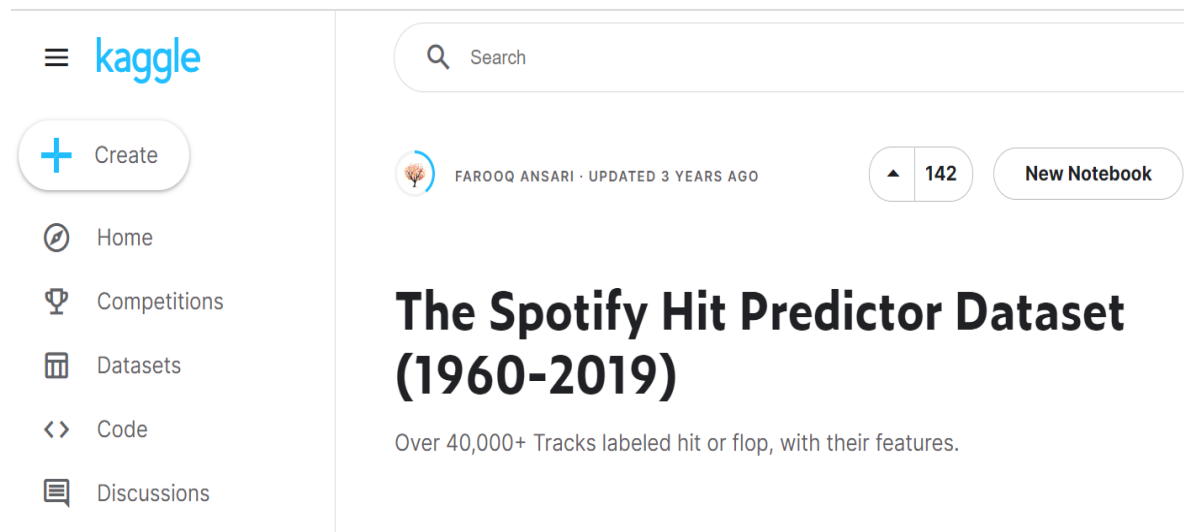
```
#import library packages
import warnings
warnings.filterwarnings("ignore")

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib Inline
from scipy.stats import ttest_ind

from sklearn.neighbors import KNeighborsClassifier
from sklearn.model_selection import GridSearchCV, train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.preprocessing import MinMaxScaler, StandardScaler
from sklearn import metrics
from sklearn.model_selection import GridSearchCV
from scipy import stats
from sklearn.model_selection import RandomizedSearchCV, KFold
from sklearn.metrics import f1_score
from sklearn.metrics import confusion_matrix
import pickle
from sklearn.metrics import classification_report
```

## Download data:

In order to develop a music recommendation system, Spotify dataset is used from Kaggle. The downloaded dataset from Kaggle is imported.



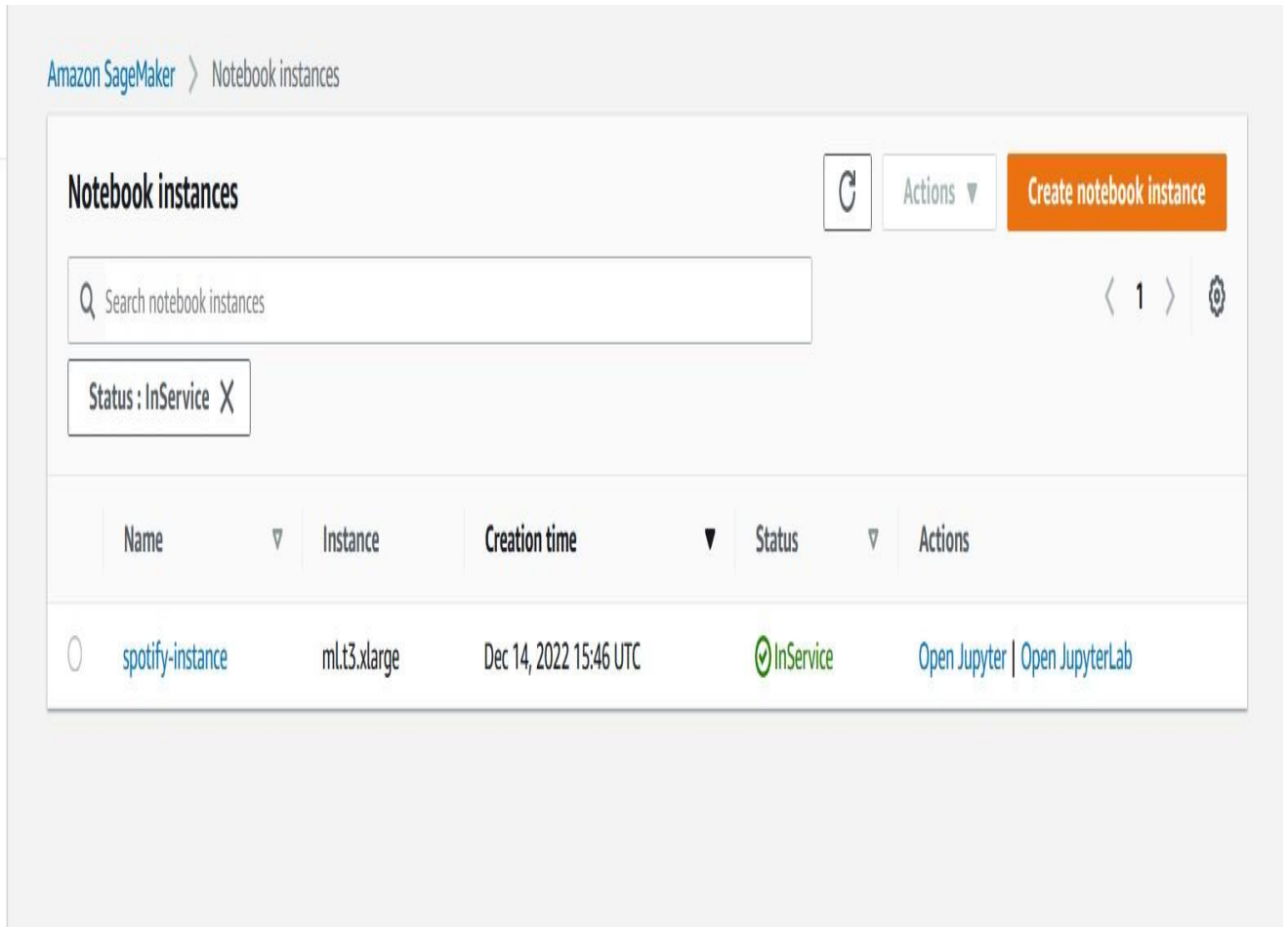
The dataset used in this study is named as “Spotify Hit Predictor Dataset”. This dataset is imported by using pandas library as a dataframe.

```
: dfs = [pd.read_csv(f'./Spotify_Hit_Predictor_Dataset2/dataset-of-{decade}0s.csv') for decade in ['6', '7', '8', '9', '0']
```

## Methodological Approach

### Amazon Sage Maker

A completely managed service for machine learning, Amazon SageMaker may be found on the Amazon website. We can swiftly and efficiently develop machine learning models with SageMaker, as well as train them, and then instantly deploy them into a production environment. It eliminates the need for us to operate servers by providing an integrated Jupyter writing notebook session, which enables quick and simple access to your data sources for the purposes of exploration and analysis. SageMaker offers versatile distributed training alternatives that may be adapted to your particular workflows.

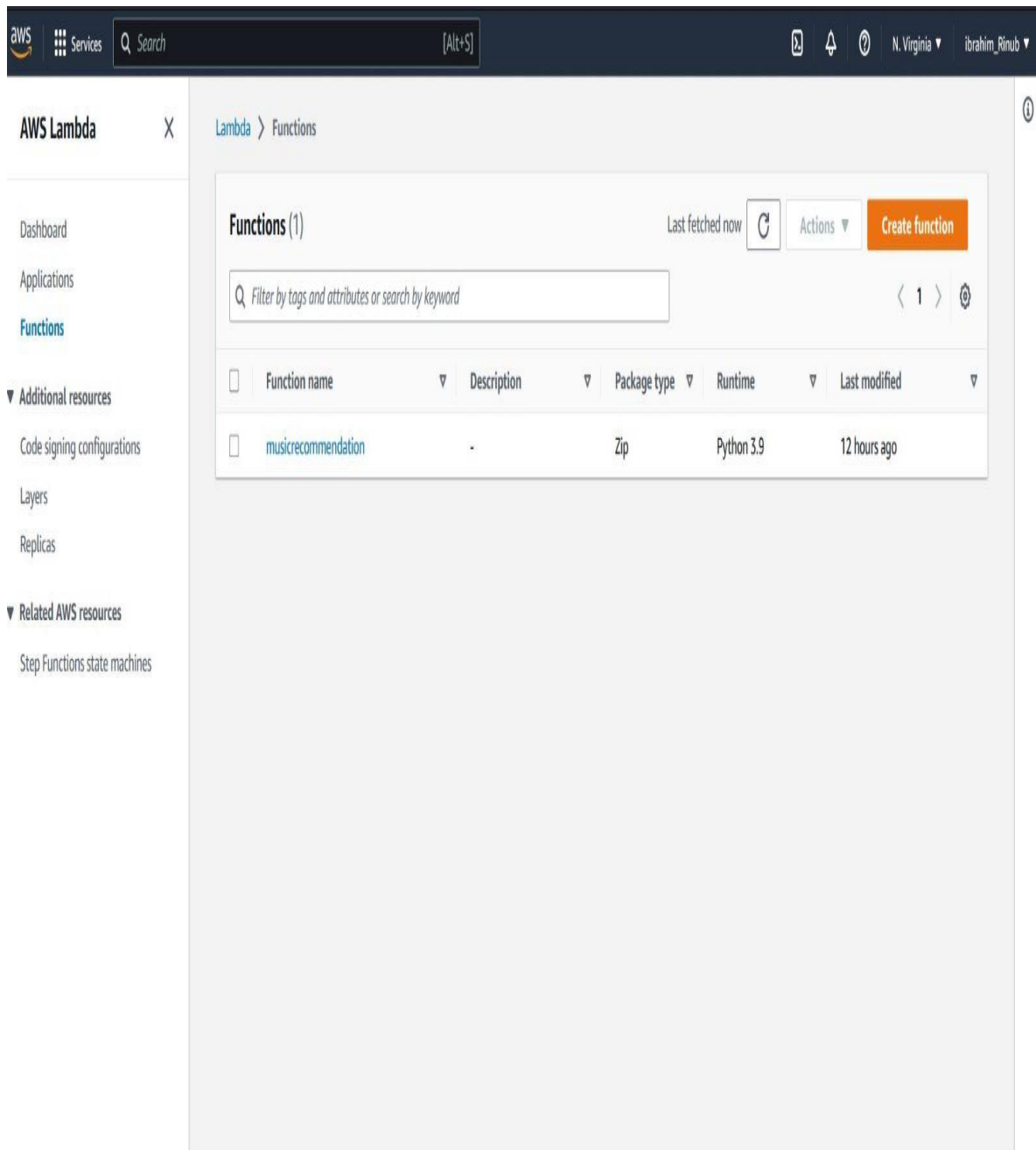


### S3 BUCKET

We have been storing everything in Amazon's S3 bucket, which is a storage solution that is known for its scalability, data availability, security, and performance. Amazon S3 allows you to store an unlimited quantity of data and access it at any time and from any location you want. The Spotify data has been saved in an S3 bucket that we control. Additionally, the lambda function will be automatically called whenever any data is added to the system.

### LAMBDA FUNCTION

A serverless computing service, AWS Lambda executes your code in response to events and takes care of the underlying resource management for you automatically. When the new data file is loaded into the system, this will cause an update to be triggered. Code is automatically executed by AWS Lambda in response to a number of events, including HTTP requests sent through Amazon API Gateway, changes made to objects stored in Amazon Simple Storage Service (Amazon S3) buckets, table updates sent through Amazon DynamoDB, and state transitions sent through AWS Step Functions. Lambda manages all of your computing resources and runs our code on high-availability infrastructure.



## CLOUD WATCH

Using cloud watch we can monitor and manage the system data and insights about the data. using this we can get the performance and operational data all in a single place. we can also place an alarm when an undefined data enters the system. all the process logs can be viewed here like whether the new data has been trained or not and the new data is deployed into the ml model or not all the insights can be gathered here.

The log events are the records of the resource being monitored. In this log events, Amazon Web Services resource's changes are described as real time stream.

### Log events

You can use the filter bar below to search for and match terms, phrases, or values in your log events. [Learn more about filter patterns](#)

Timestamp	Message
No older events at this moment. <a href="#">Retry</a>	
2022-12-14T20:51:03.737+00:00	START RequestId: 20ee6e59-c9c2-4a70-a7e2-adeb74f5137a Version: \$LATEST
2022-12-14T20:51:03.992+00:00	Training the ML model with the new dataset
2022-12-14T20:51:03.992+00:00	Train data added to S3 Bucket in _text/csv_Format
2022-12-14T20:51:03.992+00:00	The ML model has been deployed to S3 Bucket Successfully
2022-12-14T20:51:04.001+00:00	END RequestId: 20ee6e59-c9c2-4a70-a7e2-adeb74f5137a
2022-12-14T20:51:04.001+00:00	REPORT RequestId: 20ee6e59-c9c2-4a70-a7e2-adeb74f5137a Duration: 264.36 ms Billed Duration: 265 ms Memory Size: 128 MB Max Memory U...
No newer events at this moment. Auto retry paused. <a href="#">Resume</a>	

## Experimental Setup

An experimental procedure needs to be put in place in order to accurately find the recommendation system. The following elements ought to be part of the experimental setup:

1. Train the Spotify data and test the data.
2. A collection of test results that can be used to gauge how well the system is working.
3. A group of measures for gauging the effectiveness of the system
4. A procedure for recording the test results.

## Schedule

A cluster is a collection of interconnected hosts or computers that collaborate to provide middleware and applications. Every computer in a cluster is referred to as a "node." In contrast to grid computers, where each node carries out a unique work, computer clusters assign each node the same assignment. Multiple Clusters are used in Sage maker for executing the results.

## Specifications

The danceability of the song, the energy of the song, the loudness of the song, the speechiness of the song, the instrument, the liveness of the instrument, the valence of the instrument, the tempo of the instrument, the

duration of the instrument in milliseconds, the sections, by this we can separate all of the songs, and the listening patterns of the users, we can suggest according to them.

```
danceability : T_stat: 74.7901595835779 / P_value: 0.0
energy : T_stat: 36.49114636517893 / P_value: 5.987111564095718e-287
loudness : T_stat: 60.51944378914344 / P_value: 0.0
speechiness : T_stat: -8.285943324340233 / P_value: 1.206807801225774e-16
acousticness : T_stat: -51.46355964028172 / P_value: 0.0
instrumentalness : T_stat: -90.50610383659532 / P_value: 0.0
liveness : T_stat: -10.443862406965835 / P_value: 1.6824912751669193e-25
valence : T_stat: 52.603756520146916 / P_value: 0.0
tempo : T_stat: 6.622784650102702 / P_value: 3.5682852808286016e-11
duration_ms : T_stat: -15.007206361516776 / P_value: 8.980976712179482e-51
chorus_hit : T_stat: -9.419065151555587 / P_value: 4.779192361662327e-21
sections : T_stat: -12.185826684530705 / P_value: 4.235949580734437e-34
```

## Conclusion

The purpose of the research was to develop a better music recommendation model that can automatically train itself with fresh data whenever it is added to the system. In order to address the issue of an unevenly distributed dataset, the KNN was applied. The findings of the models point to a good performance on the part of the models in addressing the problem with the study, and the implementation of this has led to an improvement in the level of accuracy. The initial phase of applying models with their default settings is followed by the implementation of a KNN model to improve model performance using the mean reciprocal rank approach. This step contributes to an improvement in the overall performance of the machine learning models. It is possible to draw the conclusion, as a closing point on the research, that the research was successful in achieving its goal by using a select number of the models that performed very well and assisted the study in achieving its objectives.