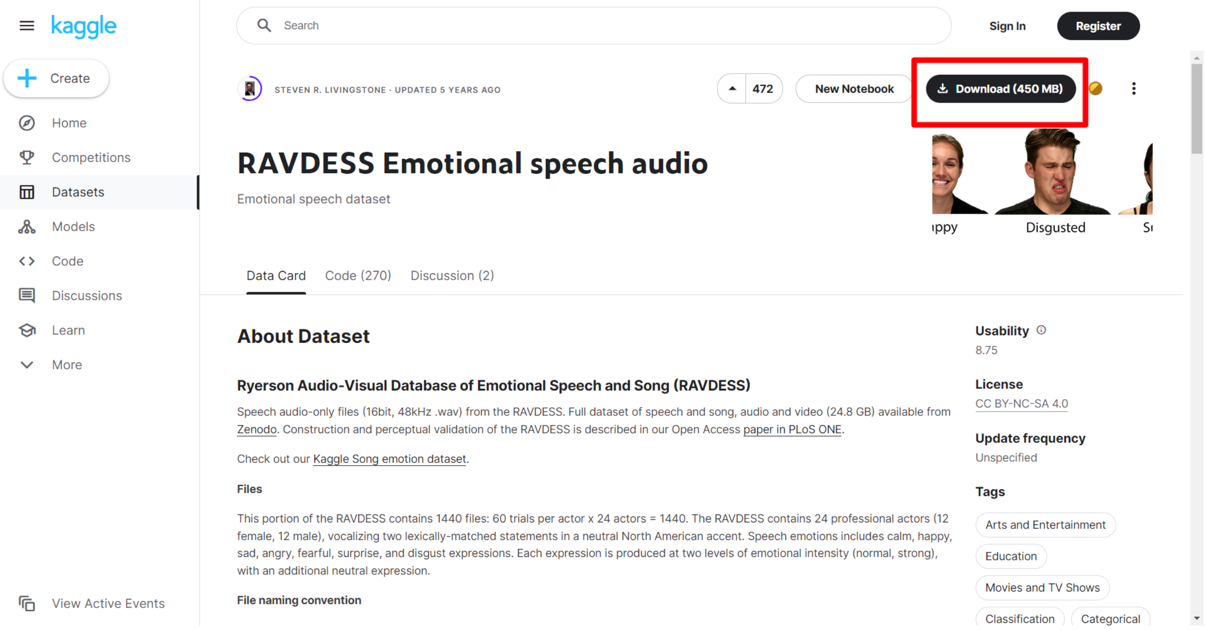
# Steps to setup the project

1. Install vscode
   https://www.youtube.com/watch?v=VvYhfj2g4Zo&pp=ygUTdnNjb2RlIGluc3RhbGxhdGlvbg%3D%3D

2. Download the dataset
   https://www.kaggle.com/datasets/uwrfkaggler/ravdess-emotional-speech-audio



RAVDESS Emotional speech audio

Emotional speech dataset

About Dataset

Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS)

Speech audio-only files (16bit, 48kHz .wav) from the RAVDESS. Full dataset of speech and song, audio and video (24.8 GB) available from Zenodo. Construction and perceptual validation of the RAVDESS is described in our Open Access paper in PLoS ONE.

Check out our Kaggle Song emotion dataset.

Files

This portion of the RAVDESS contains 1440 files: 60 trials per actor x 24 actors = 1440. The RAVDESS contains 24 professional actors (12 female, 12 male), vocalizing two lexically-matched statements in a neutral North American accent. Speech emotions includes calm, happy, sad, angry, fearful, surprise, and disgust expressions. Each expression is produced at two levels of emotional intensity (normal, strong), with an additional neutral expression.

File naming convention

Usability
8.75

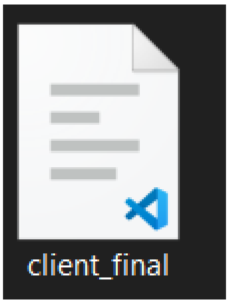
License
CC BY-NC-SA 4.0

Update frequency
Unspecified

Tags
Arts and Entertainment
Education
Movies and TV Shows
Classification
Categorical

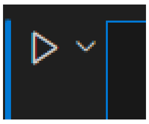
**Recommended**- Download the dataset on Desktop or Downloads

3. Now download the client_final.ipynb which was shared with you.

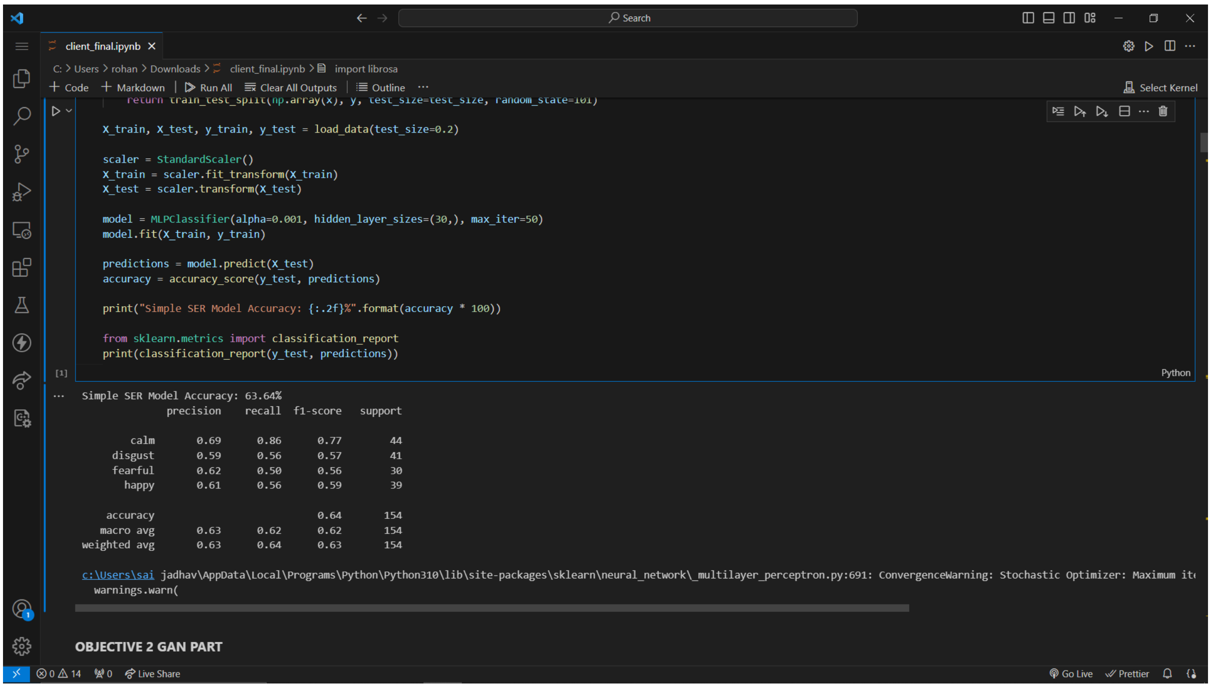


4. Double click on it to open in vscode

5. After opening the vscode, go to extensions on the left side of the window and install jupyter notebook extension https://code.visualstudio.com/docs/datascience/jupyter-notebooks
6. Refer the above document on how to run a particular block of code and run the code-



7. Keep on running each and every block and the output of a particular code block will be displayed below the code itself.



Search
client_final.ipynb
C: > Users > rohan > Downloads > client_final.ipynb > import librosa
+ Code + Markdown | Run All | Clear All Outputs | Outline ...
Select Kernel
[1]
Python

```python
    return train_test_split(np.array(x), y, test_size=test_size, random_state=101)

X_train, X_test, y_train, y_test = load_data(test_size=0.2)

scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)

model = MLPClassifier(alpha=0.001, hidden_layer_sizes=(30,), max_iter=50)
model.fit(X_train, y_train)

predictions = model.predict(X_test)
accuracy = accuracy_score(y_test, predictions)

print("Simple SER Model Accuracy: {:.2f}%".format(accuracy * 100))

from sklearn.metrics import classification_report
print(classification_report(y_test, predictions))
```

```
Simple SER Model Accuracy: 63.64%
              precision    recall  f1-score   support

        calm       0.69      0.86      0.77        44
     disgust       0.59      0.56      0.57        41
     fearful       0.62      0.50      0.56        30
       happy       0.61      0.56      0.59        39

    accuracy                           0.64       154
   macro avg       0.63      0.62      0.62       154
weighted avg       0.63      0.64      0.63       154

c:\Users\sai jadhav\AppData\Local\Programs\Python\Python310\lib\site-packages\sklearn\neural_network\_multilayer_perceptron.py:691: ConvergenceWarning: Stochastic Optimizer: Maximum it
  warnings.warn(
```

OBJECTIVE 2 GAN PART