Loading data

```
import librosa
from glob import glob
#directories of normal audios
data_dir = "D:\\signal processing\\Language Detection Dataset\\Marathi"
audio = librosa.util.find_files(data_dir , ext=['wav'])
audio
```

Extracting features from audio data

```
import librosa
from librosa import feature
import numpy as np
list_i = [
    feature.chroma_stft,
    feature.spectral_centroid,
    feature.spectral_bandwidth,
    feature.spectral_rolloff
]

list_ii = [
    feature.zero_crossing_rate
]
def feature_vector(y,sr):
    feat_vect_i = [ np.mean(funct(y,sr)) for funct in list_i]
    feat_vect_ii = [ np.mean(funct(y)) for funct in list_ii]
    feature_vector = feat_vect_i + feat_vect_ii
    return feature_vector
```

Converting features into list

```
 list = []
for file in audio:
    y , sr = librosa.load(file,sr=None,offset = 0.0,duration = 5.0)
    feature_vector = get_feature_vector(y, sr)
```

```
    list.append(feature_vector)
```

Creating csv file of features

```
import csv
output = "data.csv"
header =[
 "chroma_stft",
 "spectral_centroid",
 "spectral_bandwidth",
 "spectral_rolloff",
 "zero_crossing_rate"
]
with open(output,"+w") as f:
 csv_writer = csv.writer(f, delimiter = ",")
 csv_writer.writerow(header)
 csv_writer.writerows(norm_audios_feat)
output
```

Loading dataset

```
import pandas as pd
data= pd.read_csv('data.csv')
data


# Importing the libraries
import numpy as np
import pandas as pd
import pandas as pd
```

Loading data

```
Data1 = "C:\\Users\\Rahul\\Desktop\\language recognition\\Final Data\\English.csv"
Data2 = "C:\\Users\\Rahul\\Desktop\\language recognition\\Final Data\\hindi.csv"
Data3 = "C:\\Users\\Rahul\\Desktop\\language recognition\\Final Data\\marathi.csv"
files = [Data1,Data2,Data3]


# Combine all three CSV files using the concat method
```

```
data = pd.concat([pd.read_csv(f) for f in files])

# Export to csv

data.to_csv( "DATA.csv", index=False, encoding='utf-8-sig')

data = pd.read_csv("DATA.csv")

data
```

Modelling-

Encoding language column

```
from sklearn.preprocessing import LabelEncoder

le = LabelEncoder()

y_data = le.fit_transform(data['Lang'])

print(y_data)

X = data.iloc[:, 0:5].values

y = y_data
```

Train test split

```
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.3, random_state = 1,
shuffle=True)

print(y_train)
```

Random Forest algorithm

```
# Fitting Random Forest Regression to the dataset

# import the regressor

from sklearn.ensemble import RandomForestClassifier


 # create regressor object

classifier= RandomForestClassifier(n_estimators = 100, bootstrap=True,random_state = 1)


# fit the regressor with x and y data

classifier.fit(X_train, y_train)


y_pred = classifier.predict(X_train)
```

```
y_pred
classifier.score(X_test,y_test)
```

Recording Audio

```
import sounddevice as sd
import soundfile as sf
from tkinter import *
 def Voice_rec():
   fs = 48000

   # seconds
   duration = 5
   myrecording = sd.rec(int(duration * fs),
    samplerate=fs, channels=2)
   sd.wait()

   # Save as FLAC file at correct sampling rate
   return sf.write('Audio1.wav', myrecording, fs)
master = Tk()

Label(master, text=" Voice Recoder : "
   ).grid(row=0, sticky=W, rowspan=5)

b = Button(master, text="Start", command=Voice_rec)
b.grid(row=0, column=2, columnspan=2, rowspan=2,
   padx=5, pady=5)
mainloop()
```

Loading Audio data

```
import librosa
list = []
file="C:\\Users\\Rahul\\Desktop\\language recognition\\Audio1.wav"
 y , sr = librosa.load(file,sr=None,offset = 0.0,duration = 5.0)
feature_vector = get_feature_vector(y, sr)
list.append(feature_vector)
```

Creating Csv from features of audio

```
import csv
output = "Input_audio.csv"
header =[
 "chroma_stft",
 "spectral_centroid",
 "spectral_bandwidth",
 "spectral_rolloff",
 "zero_crossing_rate"
]
with open(norm_output,"+w") as f:
    csv_writer = csv.writer(f, delimiter = ",")
    csv_writer.writerow(header)
    csv_writer.writerows(list)

import pandas as pd
Input= pd.read_csv('Input_audio.csv')
Input
Input=data.loc[0].values.tolist()
```

Predicting Output

```
print(classifier.predict(([Input[0:5]])))
```