Team Project Report

Classification of voice

Team Name:Salt Fish

Team Members:丁宁 陈梦霞 梁海娜 王清凝 王敏

1. Introduction

1.1 Functions

Voice Gender Recognition System can Identify a voice as male or female, based upon acoustic properties of the voice and speech.

Methods for automatic gender recognition based on voice allow the person identification only by analysis of this person speech recording.

So we come up with the idea that we can create a voice gender recognition system as the prototype of a voice software, which helps distinguish male from female through voice analysis.

1.2 Method

We compare three machine-learning methods to choose the best one to distinguish which gender the voice belongs to.

The accuracy of k Nearest Neighbors、Decision Tree and Random Forest is shown below.

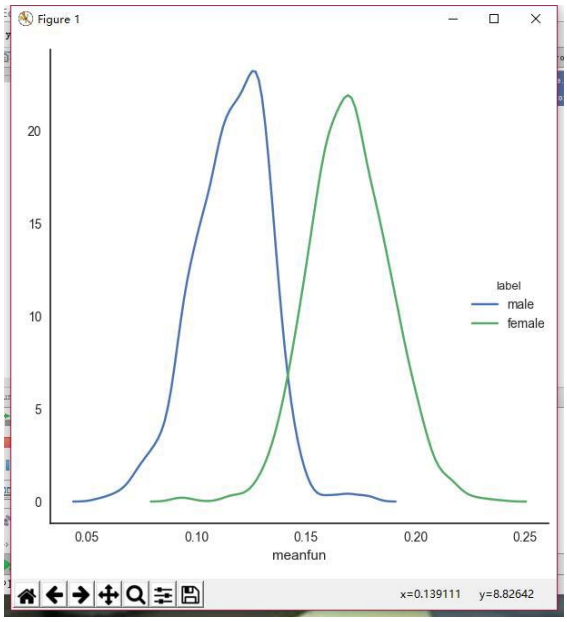
```
Score of KNN: 0.9416403785488959
Score of TREE: 0.9148264984227129
Score of FOREST: 0.9148264984227129
```

Clearly，the accuracy of KNN is the highest. So we choose the KNN method.

1.3 Key Factor

Pitch is an important feature of voice signal, determining whether the voice frequency is high or low. If we directly observe the sound wave,we can easily find out the cycle of frequency.So we choose frequency as a feature to distinguish voices gender.

We first write code to see the pitch frequency of men and women. And then get the figure showing pitch frequency of male and female with over 3000 voices from the database.



Finally the result shows that normally if frequency is less than 150Hz, the person is probably a man, and if it is more than 150Hz, the person is probably a woman.

2. Task Allocation

丁宁:presentation
陈梦霞:code
梁海娜:code
王清凝:team project report
王敏:PPT

3.Algorithm Description

3.1 Split voice data into train data and test data and get the KNN model

```
data = pd.read_csv('./voice.csv')
feature_columns = ["meanfun"]
X = data[feature_columns]
y = data['label']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=0)
model_knn = KNeighborsClassifier(n_neighbors=10)
model_knn.fit(X_train, y_train)
y_predict = model_knn.predict(X_test)
model_tree = DecisionTreeClassifier()
model_tree.fit(X_train, y_train)
y_predict = model_tree.predict(X_test)
model_forest = RandomForestClassifier(n_estimators=50)
model_forest.fit(X_train, y_train)
y_predict = model_forest.predict(X_test)
print('Score of KNN:',format(model_knn.score(X_test, y_test)))
print('Score of TREE:',format(model_tree.score(X_test, y_test)))
print('Score of FOREST:',format(model_forest.score(X_test, y_test)))
```

3.2 Set voice parameters to record format of voices

```
CHUNK = 1024
FORMAT = pyaudio.paInt16
CHANNELS = 2
RATE = 44100
RECORD_SECONDS = 5
```

3.3 Ask users to record voice, collect data and save them in a file.

```
p = pyaudio.PyAudio()

stream = p.open(format=FORMAT,
                channels=CHANNELS,
                rate=RATE,
                input=True,
                frames_per_buffer=CHUNK)

print("* recording")

frames = []

for i in range(0, int(RATE / CHUNK * RECORD_SECONDS)):
    data = stream.read(CHUNK)
    frames.append(data)

print("* done recording")

stream.stop_stream()
stream.close()
p.terminate()

wf = wave.open(WAVE_OUTPUT_FILENAME, 'wb')
wf.setnchannels(CHANNELS)
wf.setsampwidth(p.get_sample_size(FORMAT))
wf.setframerate(RATE)
wf.writeframes(b''.join(frames))
wf.close()
```

3.4 Read wave file and get parameters

```
fw = wave.open('D:/大一第二学期/商务计算导论/output1.wav','rb')
params = fw.getparams()
print(params)
nchannels, sampwidth, framerate, nframes = params[:4]
strData = fw.readframes(nframes)
waveData = np.fromstring(strData, dtype=np.int16)
waveData = waveData*1.0/max(abs(waveData))  # normalization
fw.close()
```

3.5 Get a picture of the frequency of the voice

```
time = np.arange(0, len(waveData)) * (1.0 / framerate)


pl.subplot(311)
pl.plot(time, waveData)
pl.xlabel("time (seconds)")
pl.ylabel("Amplitude")

pl.show()
```

3.6 Find out the main speaking time and draw two lines in the picture

```
a=input("please input the index1")
a=float(a)
a=framerate*a
a=int(a)
index1 = a/ framerate
index2 = (a+512) / framerate
pl.subplot(311)
pl.plot(time, waveData)
pl.plot([index1, index1], [-1, 1], 'r')
pl.plot([index2, index2], [-1, 1], 'r')
pl.xlabel("time (seconds)")
pl.ylabel("Amplitude")
```

3.7 Find out the period of the voice and input the period start point and end point

```
pl.subplot(312)
pl.plot(np.arange(512), waveData[a:a+512], 'r')
pl.xlabel("index in 1 frame")
pl.ylabel("Amplitude")
pl.show()
```

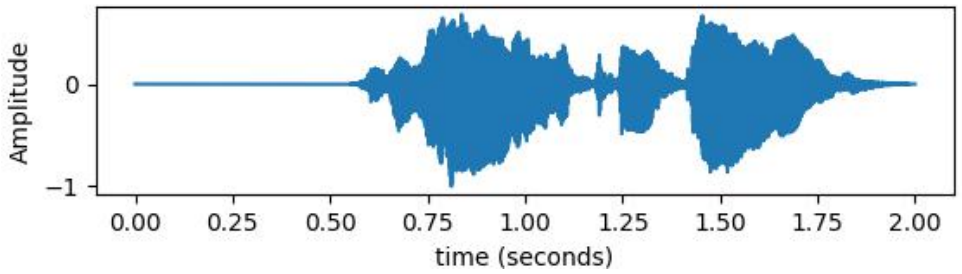
```
b=input("please input the start:")
b=float(b)
c=input("please input the finish:")
c=float(c)
pl.plot([b,b],[-1,1],'b')
pl.plot([c,c],[-1,1],'b')
print("the frequency is:",1/((c-b)*1.0/framerate),"Hz")
```

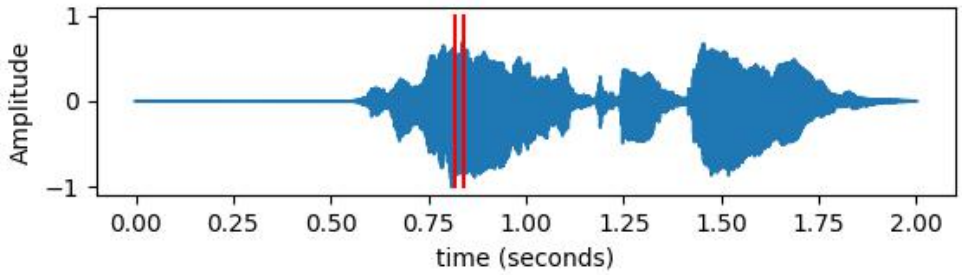
3.8 Get the gender according to the frequency

```
gender = model_knn.predict(1/(0.001*(c-b)*1.0/framerate))
print("The voice comes from a ",gender)
```

4.Testing result

4.1 Get a picture of sound wave

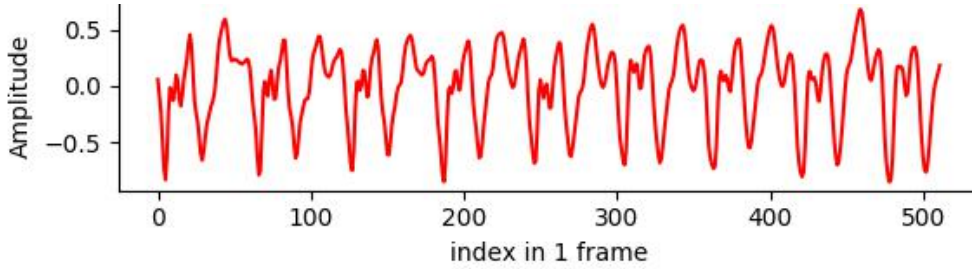




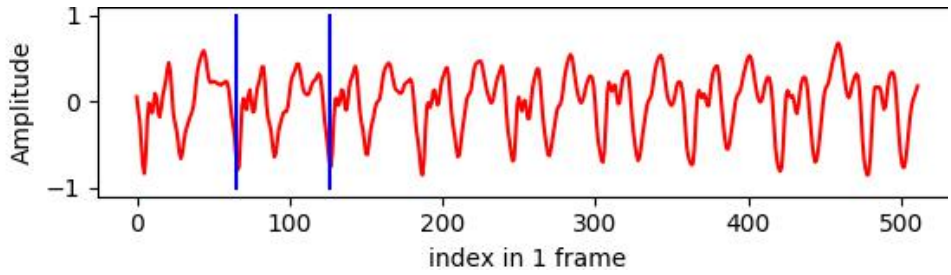
The lateral axis is time and the vertical axis is frequency.

In this picture are the changes in frequency of sound as time changes. And we take a part of the data where voice exists.

4.2 Get the larger picture of a short period



In this picture we can find a pitch, and our team choose two adjacent trough of waves to make plumb line. Then we get two numbers to do the next calculation.



According to these two numbers, we get the result

We can see the frequency of the data, then we put it into the best machine-learning method(KNN) to see the probability whether the person is male or female.

Finally it turns out that the person is a female.

5.Conclusion

5.1 Findings

KNN is the best method to distinguish which gender the voice's owner belongs to.

5.2 Reflection

Although the algorithm which takes frequency as the main feature for distinguishing male from female, the algorithm used to calculate frequency is relatively simple， ignoring complicated factors and definitely causes deviation. Since the less noisy the environment is, the more accurate the calculation of frequency will be, the environment of audio record is demanding. Besides, the voice signal is expected to be stable， at least showing obvious periodic changes. In fact, the calculation of frequency has more accurate algorithm. Nevertheless, it involves knowledge of physics, our team members choose the more feasible algorithm after comparison.