به نام خدا

Persian Speech Emotion Recognition in MATLAB

تشخیص احساسات در گفتار فارسی با استفاده از متلب

تهیه کننده : علیرضا امیری

شماره دانشجویی : 982151028

استاد راهنما : جناب آقای دکتر حنان لهراسبی

چکیده:

در این پروژه تلاش می شود با انجام تحلیل های مختلف بر روی داده های صوتی، بتوانیم "احساسات" و "جنسیت" گوینده را تشخیص بدهیم. برای این منظور، ابتدا نمونه داده هایی تهیه می کنیم و آنها را مطابق احساساتی که در آن وجود دارد، برچسب می زنیم. سپس لازم است این داده ها در پوشه هایی به تفکیک برچسب دسته بندی شوند تا بتوانیم در برنامه، از آنها استفاده کنیم. در گام بعد، این فایل به عنوان یک "مجموعه داده" یا "Dataset" برای متلب تعریف می شود. حال ما قادر هستیم تحلیل های لازم را بر روی این مجموعه داده انجام دهیم.

برای انجام این پروژه، ما از چهار تحلیل (MFCC (Mel-Frequency-Cepstral-Cooficients، Zero-Crossing-Rate، Pitch و Short-Time-Energy استفاده کرده ایم.

با اعمال این تحلیل ها بر روی تمام داده ها و ذخیره کردن آنها، می توانیم یک مدل یادگیری ماشین طراحی کنیم و آن را آموزش دهیم تا در هنگام دریافت فایل صوتی جدید، بتواند آن صوت را با مجموعه داده ی خود مقایسه و آن را با برچسب درست، شناسایی کند.

همچنین برای آموزش دادن سیستم دسته بندی کننده (Classifier)، از الگوریتم KNN (K-Nearest-Neighbors) استفاده کرده ایم.

در انتها، با انجام عملیات های Validating، میزان صحت پیش بینی های سیستم را آزمایش کرده ایم و یک بار نیز، با داده های نمونه ای که قبلا در مجموعه داده ی سیستم تعریف نشده است، آن را می آزماییم.

1. مجموعه داده (Dataset)

برای آموزش دادن و training یک مدل یادگیری ماشین، اولین چیزی که نیاز داریم یک "مجموعه داده" و "دیتاست" است. دیتاست به داده هایی اطلاق می شود که از قبل به صورت آماری گردآوری شده اند و مطابق دسته بندی های مورد نظر ما، برچسب زده شده اند. مواردی که اینجا برای ما حائز اهمیت است، اول صحت برچسب زدن داده ها و سپس، تعداد داده ها است. برای دقت بیشتر و پیش بینی های دقیق تر، به تعداد داده ی بیشتری نیاز داریم، اما بدیهی است که داده های بیشتر، منجر به حجم پردازش بیشتر می شود. بنابراین باید با توجه به هدفی که از انجام پروژه داریم، دیتاستی با تعداد متناسب تهیه کنیم.

برای انجام این پروژه، به دیتاستی نیاز داشتیم که بر حسب احساسات مختلف گوینده تقسیم شده باشد و همچنین دسته بندی آن، تفاوت جنسیت را نیز شامل شود. دیتاستی که انتخاب کردیم، شامل 3000 فایل صوتی کوتاه، با دسته بندی های { خشمگین، ناراحت، مضطرب، ترسیده، خوشحال و بی احساس } و به تفکیک جنسیت می باشد که از گفتگو ها و نمایش های رادیویی فارسی زبان استخراج شده است. این مجموعه به این دلیل انتخاب شد که گویندگان رادیو توانایی خوبی در بیان احساسات در گفتارشان دارند و از نظر، میتوانیم نسبت به صحت برچسب های هر نمونه مطمئن باشیم.

این دیتاست از لینک های موجود در قسمت منابع در انتهای گزارش قابل دانلود است.

داده های دسته بندی شده و تعدادشان در جدول زیر نمایش داده شده اند.

جنسیت و احساس	تعداد نمونه های صوتی	جنسیت و احساس	تعداد نمونه های صوتی
Female_Angry	455	Male_Angry	604
Female_Fear	22	Male_Fear	16
Female_Happy	111	Male_Happy	90
Female_Neutural	284	Male_Neutural	744
Female_Sad	271	Male_Sad	178
Female_Woried	120	Male_Woried	105

حال با استفاده از دستور audioDatastore، این دیتاست را برای متلب تعریف می کنیم.

```
ADS = audioDatastore('Dataset location);
```

برای طراحی یک مدل یادگیری ماشین، لازم است تا بخشی از دیتاست صرف آموزش مدل، و بخش دیگری برای تست کردن مدل استفاده شود. برای این منظور، دیتاست را به دو بخش ADSTrain و ADSTest به نسب 80 درصد و 20 درصد تقسیم می کنیم.

```
[ADSTrain,ADSTest] = splitEachLabel(ADS,0.8)
```

2. استخراج ویژگی ها (Feature Extraction)

در ادامه، باید ویژگی های مورد نظرمان را از داده های ADSTrain استخراج کنیم که این کار به وسیله‌ی دستور audioFeatureExtractor انجام می شود.

```
afe = audioFeatureExtractor(SampleRate=fs , Window=hamming(windowLength,"periodic"),
OverlapLength=overlapLength,
zerocrossrate=true,shortTimeEnergy=true,pitch=true,mfcc=true)
```

در اینجا، ما از چهار تحلیل MFCC، Zero-Crossing-Rate، Pitch و Short-Time-Energy استفاده کرده ایم که در بخش های زیر هر یک را به طور خلاصه توضیح می دهیم.

MFCC 2.1

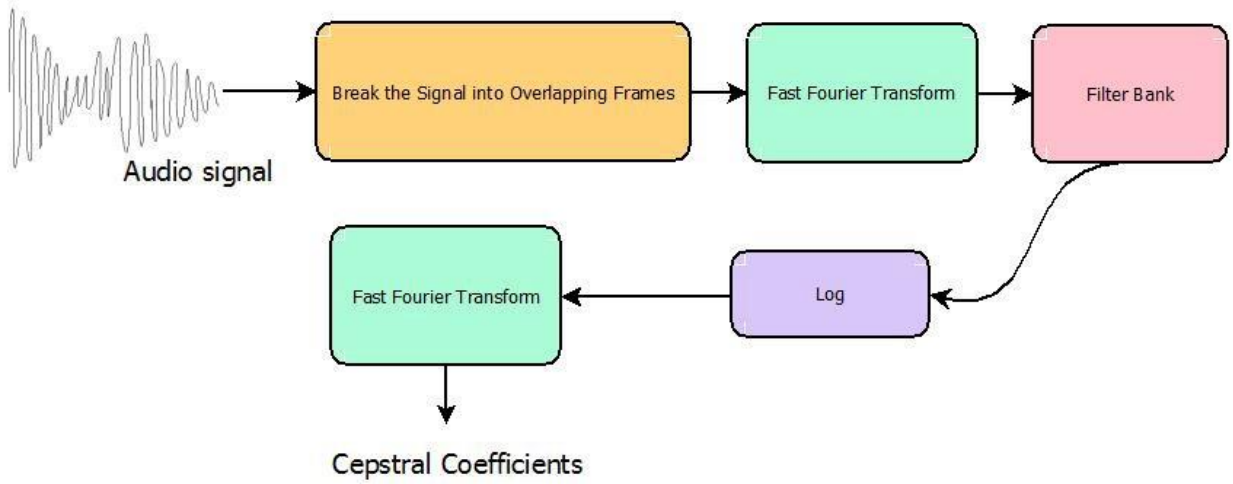
کاری که این تحلیل انجام می دهد، مشابه تبدیل فوریه است. به این صورت که ابتدا داده های صوتی را از حوزه ی زمان، به حوزه ی فرکانس می برد. اما از آنجا که نیاز داریم محتوای فرکانسی موجود را به صورت لحظه ای و یا در فاصله های زمانی کوتاه در اختیار داشته باشیم، از Short Time Fourier Transform استفاده می کنیم.

نکته ی دیگر درباره ی این روش این است که MFCC داده های صوتی را متناسب با میزان شنوایی انسان پردازش می کند. می دانیم که گوش انسان، نسبت به فرکانس های پایین تر حساس تر است؛

به طور مثال، ما تفاوت بین صداهایی با فرکانس 500 و 700 را بیشتر از صداهایی با فرکانس 1500 و 1700 متوجه می شویم.

در اینجا نیز، با اعمال یک filter bank که نسبت به فرکانس های بالاتر، حساسیت کمتری دارد، سیگنال صوتی مان را فیلتر می کنیم و سپس شدت آن را نیز با مقیاس لگاریتمی بیان می کنیم. در گام اخر، بار دیگر از محتوای فرکانسی موجود تبدیل فوریه یا تبدیل کسینوس گرفته و آن را به حوزه ای به جز زمان و فرکانس می بریم، که به این حوزه Quefrency و به محتوای آن Cepstrum گفته می شود.

فرایند گفته شده در نمودار زیر قابل مشاهده است.



Short-Time-Energy 2.2

هدف از به کار گیری این معیار، حذف کردن قسمت هایی از فایل صوتی است که شامل هیچ گفتاری نیستند. این قسمت ها، دارای انرژی کمی هستند و بنابراین می توانیم با اندازه گیری انرژی در بازه های کوتاه، آنها را شناسایی و حذف کنیم.

Pitch 2.3

این ویژگی، نشان دهنده ی فرکانس صدا است و می توانیم از آن در تشخیص جنسیت گوینده استفاده کنیم. اگرچه، در اختیار داشتن این ویژگی از تمام دیتاست، در تشخیص احساسات نیز کمک بزرگی محسوب می شود.

Zero-Crossing-Rate 2.4

معیار ZCR نشان دهنده ی نرخ تغییرعلامت شکل موج از مثبت به منفی و از منفی به مثبت است. هر چه مقدار این معیار کمتر باشد، بدان معنی است که آوای مشخص تری داریم و هر چه این مقدار بیشتر شود، تشخیص آوا سخت تر شده و صدا به نویز شباهت پیدا می کند. از این معیار برای جداسازی بخش هایی از وویس استفاده می کنیم که آوای مشخصی ندارند. به عنوان مثال، حرف "ت" حاوی فرکانس مشخصی

نیست و بنابراین zcr بالایی نیز دارد که ما برای داشتن یک دیتاست شفاف و مشخص، نیاز داریم این قسمت ها را از وویس حذف کنیم.

در کد زیر، ابتدا با دو معیار zcr و short-time-energy، بخش های اضافی هر وویس را حذف کرده و سپس feature های آن وویس را استخراج کرده و برچسب می زنیم و این داده ها را در ماتریس های Features و labels ذخیره می کنیم. این عملیات تا زمانی که داده ای برای پردازش باقی مانده باشد ادامه می یابد.

```
while hasdata(ADSTrain)
[audioIn,dsInfo] = read(ADSTrain);
feat = extract(afe,audioIn);
   isSpeech = feat(:,featureMap.shortTimeEnergy) > energyThreshold;
   isVoiced = feat(:,featureMap.zerocrossrate) < zcrThreshold;
   voicedSpeech = isSpeech & isVoiced;
feat(~voicedSpeech,:) = [];
    feat(:,[featureMap.zerocrossrate,featureMap.shortTimeEnergy]) = [];
    label = repelem(dsInfo.Label,size(feat,1));
    Features = [Features;feat];
    labels = [labels,label];
    dsInfo.FileName
end
```

در انتهای این بخش، نیاز است داده ها را به فرم نرمال و استاندارد بدیل کنیم که این کار به صورت زیر انجام می شود.

```
M = mean(Features,1)
S = std(Features,[],1)
Features = (Features-M)./S
```

3.آموزش و اعتبارسنجی (Training and Validationg)

Training 3.1

در این قسمت، با در اختیار داشتن ماتریس های ویژگی و برچسب ها (features and labels) اقدام به آموزش دادن مدل یادگیری ماشین خود می کنیم. روش های مختلفی برای این کار وجود دارد، که ما از روش KNN (K-Nearest-Neighbors) استفاده می کنیم. در متلب، این کار به وسیله دستور fitcknn انجام می شود.

```
trainedClassifier = fitcknn(Features,labels, ...
    Distance="euclidean", ...
    NumNeighbors=5, ...
    DistanceWeight="squaredinverse", ...
    Standardize=false, ...
    ClassNames=unique(labels));
```

Validating 3.2

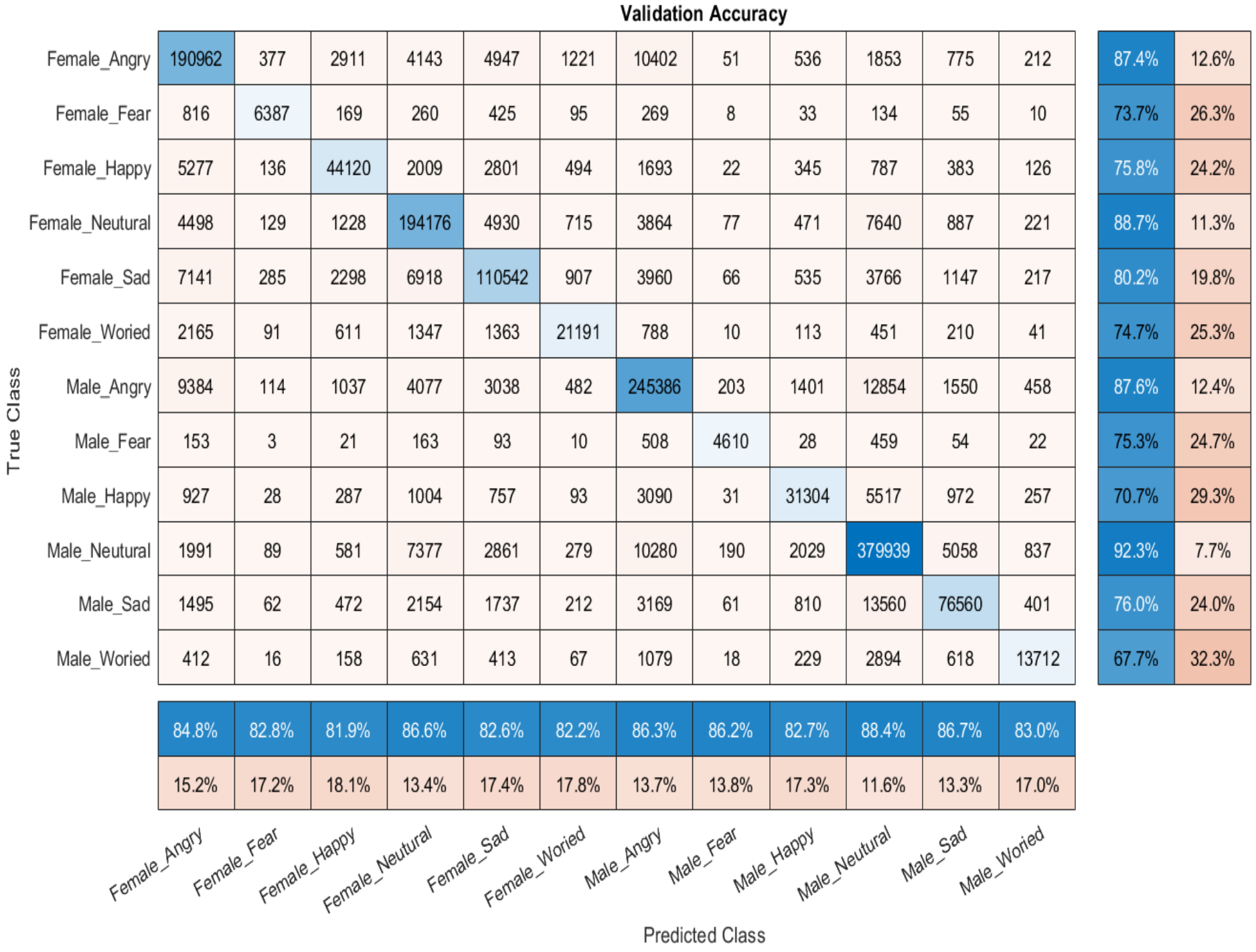
برای اعتبارسنجی، از روش K-Fold استفاده کرده ایم. به این صورت که دیتاست آموزشی (ADSTrain) را به k قسمت تقسیم می کنیم و در هر مرحله، یکی از این بخش ها را به عنوان داده ی آزمایشی و مابقی را به عنوان داده ی آموزشی تعریف می کنیم. در نتیجه، پس از k بار تکرار این فرایند، میتوانیم متوجه شویم که پیش بینی های مدل ما تا چه اندازه معتبر است و تشخیص درستی می دهد. در اینجا، k=5 در نظر گرفته شده است.

```
k = 5
group = labels
c = cvpartition(group,KFold=k); % 5-fold stratified cross validation
partitionedModel = crossval(trainedClassifier,CVPartition=c)
validationAccuracy = 1 - kfoldLoss(partitionedModel,LossFun="ClassifError")
fprintf('\nValidation accuracy = %.2f%%\n', validationAccuracy*100)
validationPredictions = kfoldPredict(partitionedModel)
```

برای این مدل، درصد اعتبار برابر با 86.03 محاسبه شده است.

همچنین در جدول زیر، نتایج حدس های درست و غلط سیستم را مشاهده می کنیم.

```
figure(Units="normalized",Position=[0.4 0.4 0.4 0.4])

confusionchart(labels,validationPredictions,title="Validation Accuracy", ...

    ColumnSummary="column-normalized",RowSummary="row-normalized");
```


Validation Accuracy
True Class
Female_Angry 190962 377 2911 4143 4947 1221 10402 51 536 1853 775 212 87.4% 12.6%
Female_Fear 816 6387 169 260 425 95 269 8 33 134 55 10 73.7% 26.3%
Female_Happy 5277 136 44120 2009 2801 494 1693 22 345 787 383 126 75.8% 24.2%
Female_Neutural 4498 129 1228 194176 4930 715 3864 77 471 7640 887 221 88.7% 11.3%
Female_Sad 7141 285 2298 6918 110542 907 3960 66 535 3766 1147 217 80.2% 19.8%
Female_Woried 2165 91 611 1347 1363 21191 788 10 113 451 210 41 74.7% 25.3%
Male_Angry 9384 114 1037 4077 3038 482 245386 203 1401 12854 1550 458 87.6% 12.4%
Male_Fear 153 3 21 163 93 10 508 4610 28 459 54 22 75.3% 24.7%
Male_Happy 927 28 287 1004 757 93 3090 31 31304 5517 972 257 70.7% 29.3%
Male_Neutural 1991 89 581 7377 2861 279 10280 190 2029 379939 5058 837 92.3% 7.7%
Male_Sad 1495 62 472 2154 1737 212 3169 61 810 13560 76560 401 76.0% 24.0%
Male_Woried 412 16 158 631 413 67 1079 18 229 2894 618 13712 67.7% 32.3%
84.8% 82.8% 81.9% 86.6% 82.6% 82.2% 86.3% 86.2% 82.7% 88.4% 86.7% 83.0%
15.2% 17.2% 18.1% 13.4% 17.4% 17.8% 13.7% 13.8% 17.3% 11.6% 13.3% 17.0%
Female_Angry
Female_Fear
Female_Happy
Female_Neutural
Female_Sad
Female_Woried
Male_Angry
Male_Fear
Male_Happy
Male_Neutural
Male_Sad
Male_Woried
Predicted Class

هر سطر این جدول، مربوط به داده های صوتی است که توسط برچسب آن مشخص شده و مورد آزمایش قرار گرفته است. داده هایی که بر روی قطر اصلی قرار دارند تعداد حدس های درست، و سایر درایه ها تعداد حدس های غلط هستند. در انتها، نسبت صحیح بودن پیش بینی های سیستم به صورت درصد برای هر برچسب نمایش داده شده است.

4.آزمایش و تست (Test)

در این مرحله، عملکرد سیستم را بر روی داده هایی که در دیتاست آن تعریف نشده اند بررسی میکنیم.
برای این منظور، ابتدا مطابق آنچه که در مرحله ی اول توضیح داده شد، 20 درصد دیتاست را برای تست (ADSTest) جدا می کنیم و فرایند آموزش و اعتبارسنجی را با 80 باقی مانده ی دیتاست (ADSTrain) انجام می دهیم . در این مرحله، داده های تست را به عنوان ورودی به سیستم می دهیم و با در اختیار داشتن مدل دسته بندی کننده (trainedClassifier)، اقدام به پیش بینی برچسب هر یک از نمونه های تست می کنیم.

این کار، مطابق فرایند زیر صورت می گیرد:

```
% Extract features from test samples and preproccess it...

features_test = [];
labels_test = [];
numVectorsPerFile = [];
while hasdata(ADSTest)
    [audioIn,dsInfo] = read(ADSTest);

    feat = extract(afe,audioIn);

    isSpeech = feat(:,featureMap.shortTimeEnergy) > energyThreshold;
    isVoiced = feat(:,featureMap.zerocrossrate) < zcrThreshold;

    voicedSpeech = isSpeech & isVoiced;

    feat(~voicedSpeech,:) = [];
    numVec = size(feat,1);
    feat(:,[featureMap.zerocrossrate,featureMap.shortTimeEnergy]) = [];

    label = repelem(dsInfo.Label,numVec);

    numVectorsPerFile = [numVectorsPerFile,numVec];
    features_test = [features_test;feat];
    labels_test = [labels_test,label];
    dsInfo.FileName
end
features_test = (features_test-M)./S;
```

ابتدا به وسیله ی دستور read، داده ها را یک به یک از ADSTest می خوانیم و ویژگی های (features) مورد نظر خود را به وسیله ی دستور extract ، از آن استخراج می کنیم. سپس مطابق آنچه که قبلا توضیح داده شد، بخش هایی از وویس که شامل گفتار و آوا نمی شوند را حذف می کنیم و ویژگی های باقی مانده را در ماتریس features_test ذخیره می کنیم. سپس، برچسب هر داده را نیز در ماتریس labels_test ذخیره می کنیم. در اینجا لازم است توضیح داده شود که این برچسب ها، در طی فرایند پیش بینی استفاده نمی شوند و سیستم، تنها با استفاده از ماتریس ویژگی های استخراج شده اقدام به حدس برچسب مورد نظر می کند، اما برای بررسی صحت حدس های سیستم، نیاز است تا برچسب های صحیح هر نمونه را نیز در اختیار داشته باشیم.

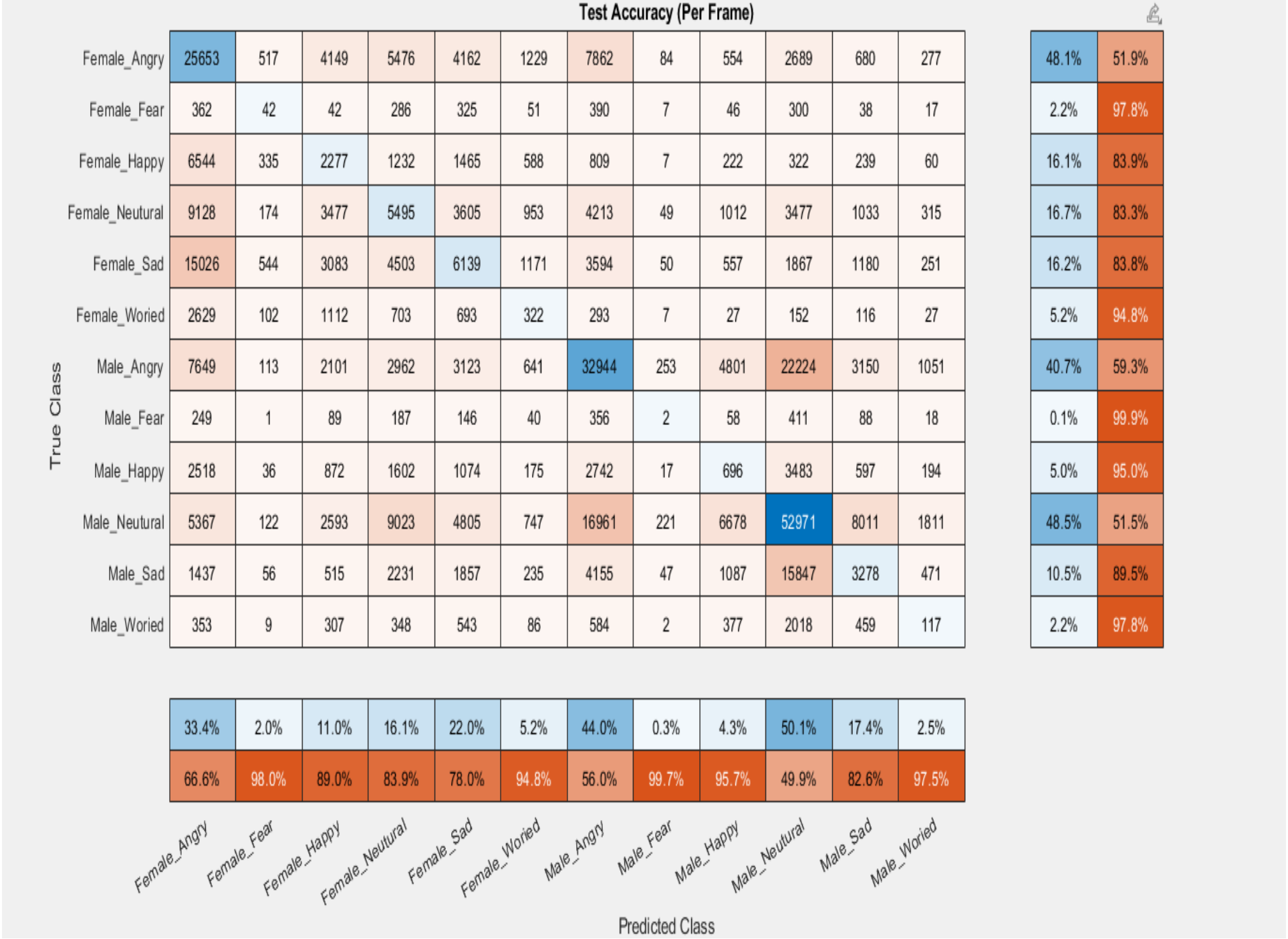
در نهایت، ماتریس ویژگی ها را به فرم نرمال تبدیل می کنیم.

```
%predict label according to trainedClassifier...

prediction = predict(trainedClassifier,features_test);
prediction = categorical(string(prediction));
```

در این قسمت، به وسیله ی دستور predict، برچسب متناسب با ویژگی هر نمونه پیش بینی میشود و در نهایت، آن را به فرم دسته بندی شده در ماتریس prediction ذخیره می کنیم.

```
% plot the result table...

figure(Units="normalized",Position=[0.4 0.4 0.4 0.4])
confusionchart(labels_test(:),prediction,title="Test Accuracy (Per Frame)", ...
    ColumnSummary="column-normalized",RowSummary="row-normalized");
```

در نهایت، نتیجه ی این تست را به صورت جدول زیر نمایش می دهیم.


Test Accuracy (Per Frame)
True Class
Female_Angry 25653 517 4149 5476 4162 1229 7862 84 554 2689 680 277 48.1% 51.9%
Female_Fear 362 42 42 286 325 51 390 7 46 300 38 17 2.2% 97.8%
Female_Happy 6544 335 2277 1232 1465 588 809 7 222 322 239 60 16.1% 83.9%
Female_Neutural 9128 174 3477 5495 3605 953 4213 49 1012 3477 1033 315 16.7% 83.3%
Female_Sad 15026 544 3083 4503 6139 1171 3594 50 557 1867 1180 251 16.2% 83.8%
Female_Woried 2629 102 1112 703 693 322 293 7 27 152 116 27 5.2% 94.8%
Male_Angry 7649 113 2101 2962 3123 641 32944 253 4801 22224 3150 1051 40.7% 59.3%
Male_Fear 249 1 89 187 146 40 356 2 58 411 88 18 0.1% 99.9%
Male_Happy 2518 36 872 1602 1074 175 2742 17 696 3483 597 194 5.0% 95.0%
Male_Neutural 5367 122 2593 9023 4805 747 16961 221 6678 52971 8011 1811 48.5% 51.5%
Male_Sad 1437 56 515 2231 1857 235 4155 47 1087 15847 3278 471 10.5% 89.5%
Male_Woried 353 9 307 348 543 86 584 2 377 2018 459 117 2.2% 97.8%
33.4% 2.0% 11.0% 16.1% 22.0% 5.2% 44.0% 0.3% 4.3% 50.1% 17.4% 2.5%
66.6% 98.0% 89.0% 83.9% 78.0% 94.8% 56.0% 99.7% 95.7% 49.9% 82.6% 97.5%
Female_Angry
Female_Fear
Female_Happy
Female_Neutural
Female_Sad
Female_Woried
Male_Angry
Male_Fear
Male_Happy
Male_Neutural
Male_Sad
Male_Woried
Predicted Class

5. نتیجه گیری

در این پروژه با در اختیار داشتن دیتاستی از داده های صوتی به تفکیک احساسات و جنسیت، یک مدل یادگیری ماشین را آموزش دادیم و از آن برای پیش بینی برچسب برای داده های جدید استفاده کردیم.

اما با توجه به نتایجی که از مرحله ی اعتبارسنجی و تست بدست آوردیم، می توان نتیجه گرفت که مجموعه ی چهار روش یاد شده برای تحلیل داده های صوتی و همچنین نحوه ی آموزش دادن مدل یادگیری ماشین (KNN) روش های مناسبی هستند و این امکان برای ما هست که با استفاده از آنها بتوانیم با درصد خوبی (86.03%) برچسب ها را حدس بزنیم. اما با دقت در نتایج تست، و همچنین با داشتن نگاهی به تعداد داده ها از برچسب های مختلف در دیتاست، می توان به این نتیجه رسید که دیتاست استفاده شده دارای صحت کافی بوده، اما از نظر تعداد و توازن، مناسب نبوده است.

در این تست، برچسب های "**Female_Angry**"،"**Male_Angry**" و "**Male_Neutural**" با احتمال 40-50 درصد درست حدس زده شده اند و این سه برچسب، بیشترین فراوانی را در دیتاست و با تعداد 450-750 نمونه داشته اند، در حالی که دیگر برچسب ها دارای تعداد نمونه ی کمتری بوده اند و با احتمال کمی، به درستی پیش بینی شده اند. از این اطلاعات، می توان نتیجه گیری کرد که برای به دست آوردن نتیجه‌ای رضایت بخش و با احتمال درستی بیش از 90%، به حدود 1500 داده برای هر برچسب در دیتاست نیاز خواهیم داشت.

6. منابع

1. **speaker-identification-using-pitch-and-mfcc:**
   https://www.mathworks.com/help/audio/ug/speaker-identification-using-pitch-and-mfcc.html?s_tid=mwa_osa_a
2. **cross-validation-machine-learning :**
   https://www.geeksforgeeks.org/cross-validation-machine-learning/
3. MATLAB Master Class Tutorial: Go from Beginner to Expert | Udemy
4. **Female Dataset link :**
   https://uc6ff32f028518ec7fe90228acb4.dl.dropboxusercontent.com/cd/0/get/BpFCFXp3gte350Ys6CiscrLawsI-3ig94O6AP9JMyBpN53E1SCyhSbbPdu8UCq_zRiqKYzlcVS4E4mrou6aIxwUrnuZnpPF9qSkggulYFvrkN6cNGEivJxCJwwuE5Hmi2w_zzHU59niKggUo3TK_t9px_DIDl4_Dx3HTCibF-lf5xLsqU3i2xReZrmp7Vzo_mnk/file?_download_id=418040282494086372957959811914240274783180970243902928539863576929&_notify_domain=www.dropbox.com&dl=1
5. **Male Dataset link :**
   https://uce0221ea80893c58c1ef7019c86.dl.dropboxusercontent.com/cd/0/get/BpGAjpasmfCsHEMfLEg76RQ7BnU5c2VcDxsTGRPcTuwnzbUu6QgoB6TgOvVk03Xcp1ZNnlYax1w-fO4EL0QAZO4Dbolt_Zb3UE2L2XMexSWyiRBKHs5uxBT1E56obE85nohrtrIbDDnKae2_BFev1FlcnkIdiH3vQCJAVwBJ3GnckCn6W6kB8HTFOSsPsV6D8-o/file?_download_id=26154154745631675601779427621408889958038865164516975312004024696&_notify_domain=www.dropbox.com&dl=1
6. **Emotion recognition from speech: a review:**
   https://doi.org/10.1007/s10772-011-9125-1
7. **Speaker Accent Recognition Using Machine Learning Algorithms :**
   https://ieeexplore.ieee.org/abstract/document/9259902