University of Pittsburgh

Qeexo Challenge

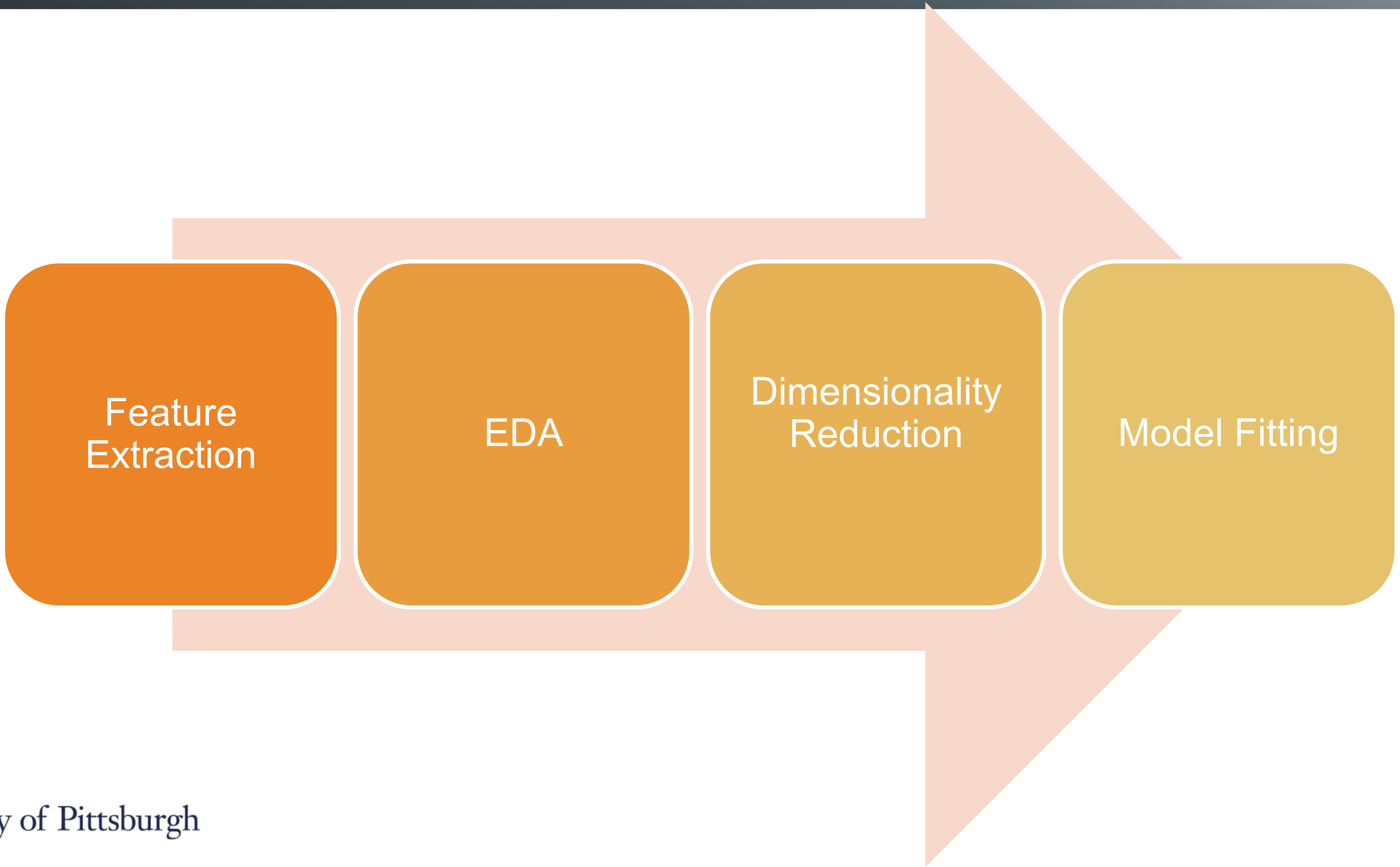
Xin Xu

Department of Industrial Engineering
University of Pittsburgh

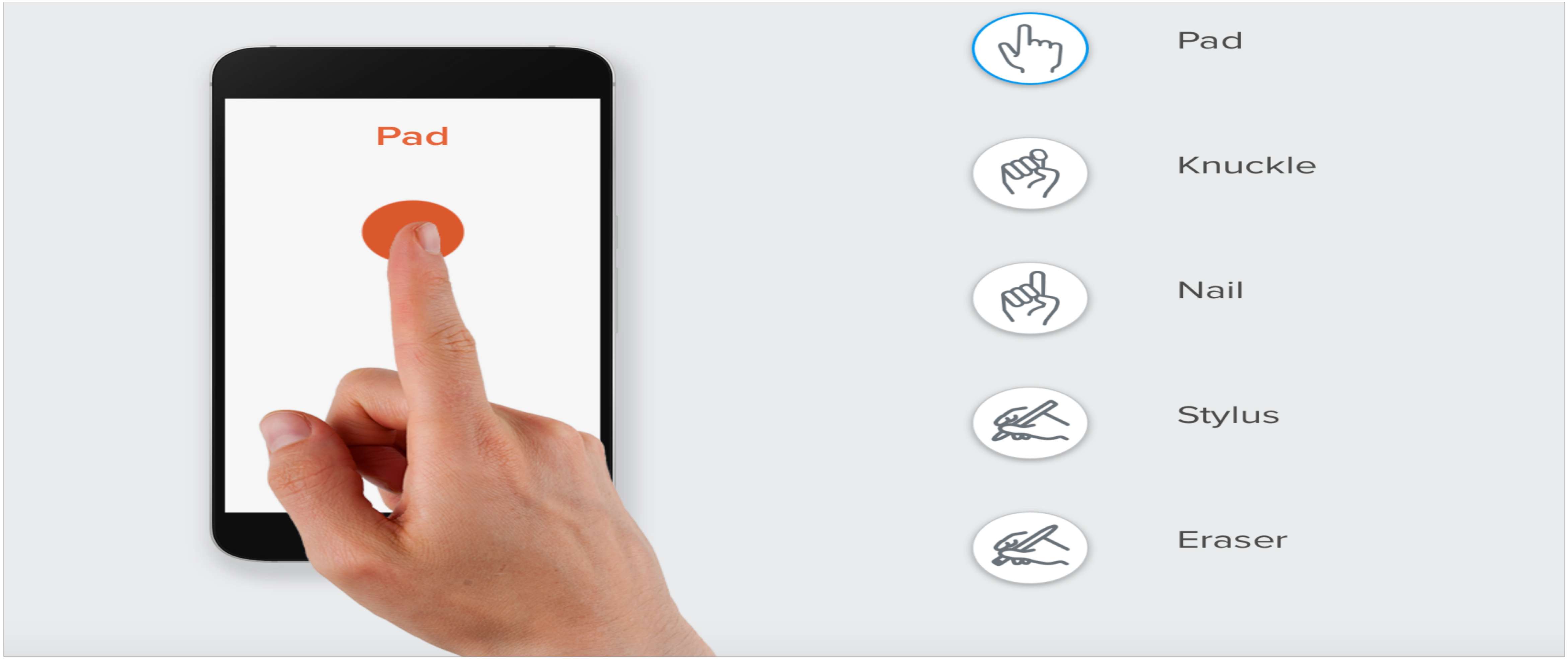
Qeexo Challenge

**Make smartphones smarter:
Pad? Knuckle?**

FingerSense



Motivation



Dataset

❖ Normal Features

- Event location(x, y)
- Touch information features(touch area size, pressure, etc)
- Surface type(hand/table)

❖ Audio Features

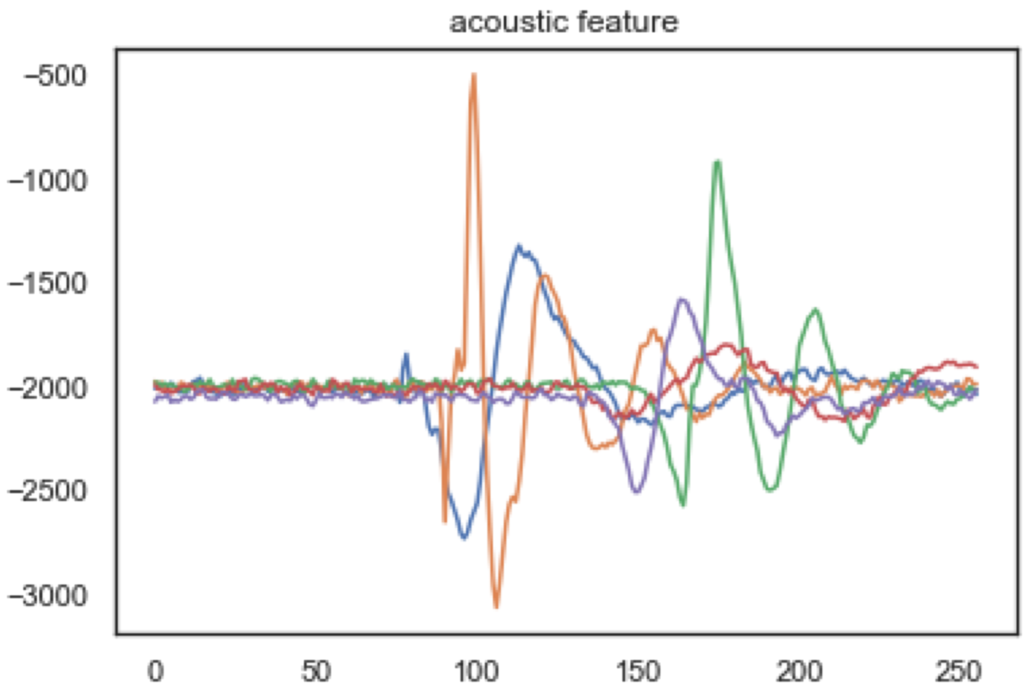
- **Audio file**

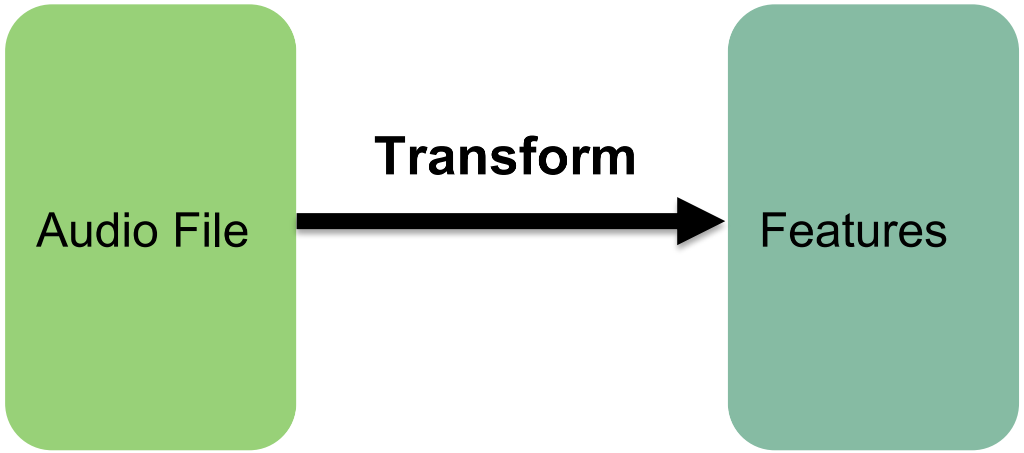
Normal Features

	x	y	major	minor	pressure	orientation	HoldingStatus
count	20659.000000	20659.000000	20659.000000	20659.000000	20659.0	20659.0	20659.000000
mean	541.317440	940.129145	4.578586	4.223099	0.0	-1.0	0.420737
std	289.368943	502.680342	1.013873	1.006628	0.0	0.0	0.493689
min	0.000000	0.000000	2.000000	2.000000	0.0	-1.0	0.000000
25%	290.000000	497.000000	4.000000	4.000000	0.0	-1.0	0.000000
50%	542.000000	938.000000	4.000000	4.000000	0.0	-1.0	0.000000
75%	791.500000	1379.000000	5.000000	5.000000	0.0	-1.0	1.000000
max	1079.000000	1919.000000	11.000000	10.000000	0.0	-1.0	1.000000

There are 7 normal features. Are they all useful?

Acoustic Features





❖ **Which method to use? MFCC? Or Just FFT?**

Features Selection

- **MFCC**(Mel-frequency cepstral coefficients)
  - Fourier transform, Map onto Mel scale, Take logs, Use DCT(discrete cosine transform)
  - Commonly used in speech recognition

- **FTT**(Fast Fourier Transform)
  - Extract features by some rules

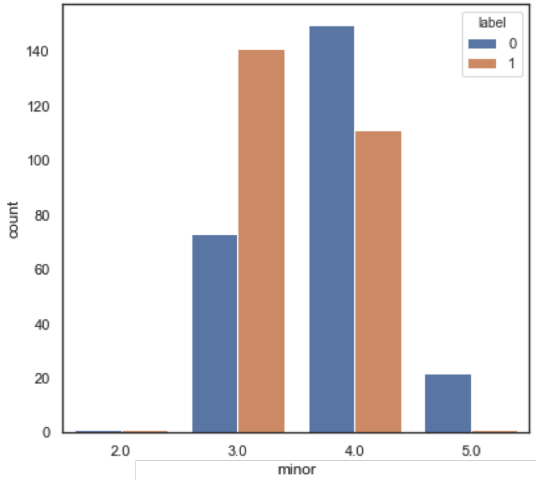
Features Selection

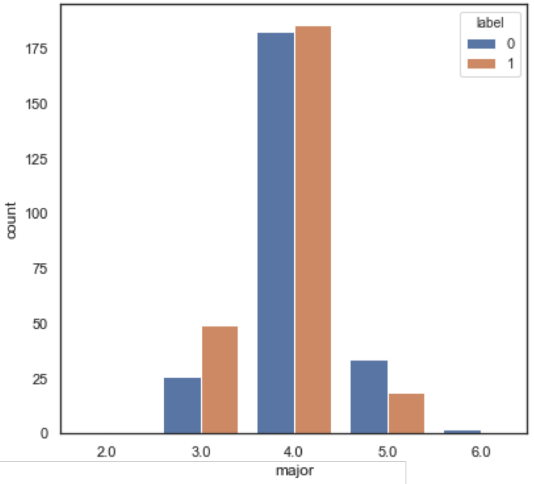
After fiting the models by using different features extraction methods, we found out direct FFT is a little bit better.

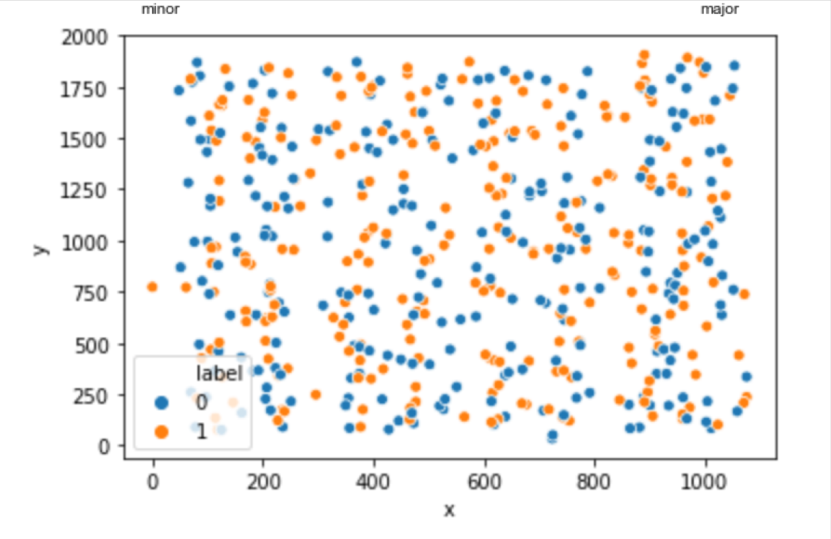
Why?

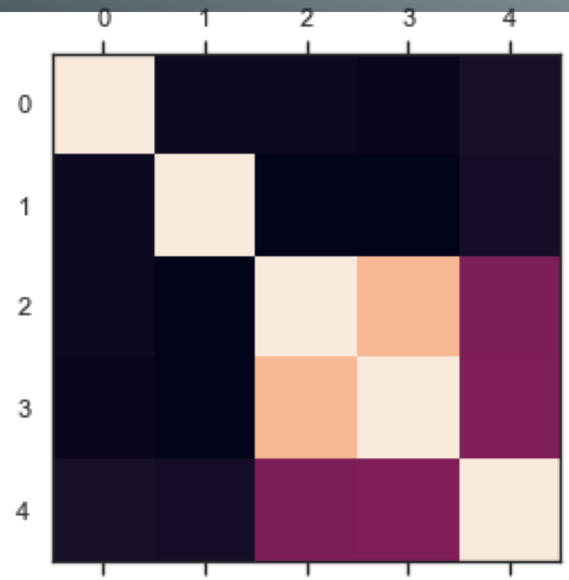
Intuition:
In MFCC, after doing Fourier Transform, it would map the spectrum onto the Mel scale. The Mel scale is good for human listeners do speech recognition. We don’t need this here. This map of Mel scale may lose some potentially useful information.

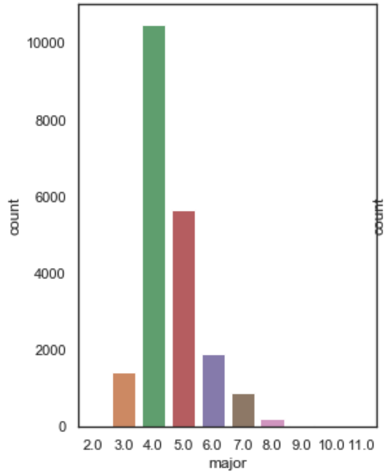
EDA

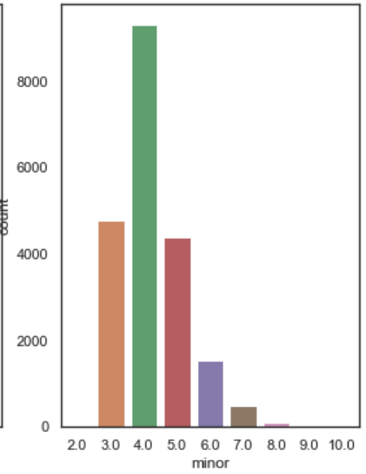

label
0
1
count
0
20
40
60
80
100
120
140
2.0
3.0
4.0
5.0
minor

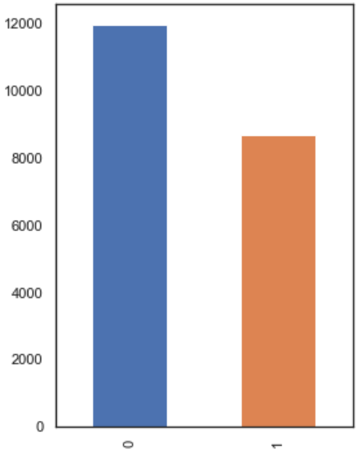

label
0
1
count
0
25
50
75
100
125
150
175
2.0
3.0
4.0
5.0
6.0
major


y
0
250
500
750
1000
1250
1500
1750
2000
x
0
200
400
600
800
1000
label
0
1


0
1
2
3
4


count
0
2000
4000
6000
8000
10000
2.0 3.0 4.0 5.0 6.0 7.0 8.0 9.0 10.0 11.0
major

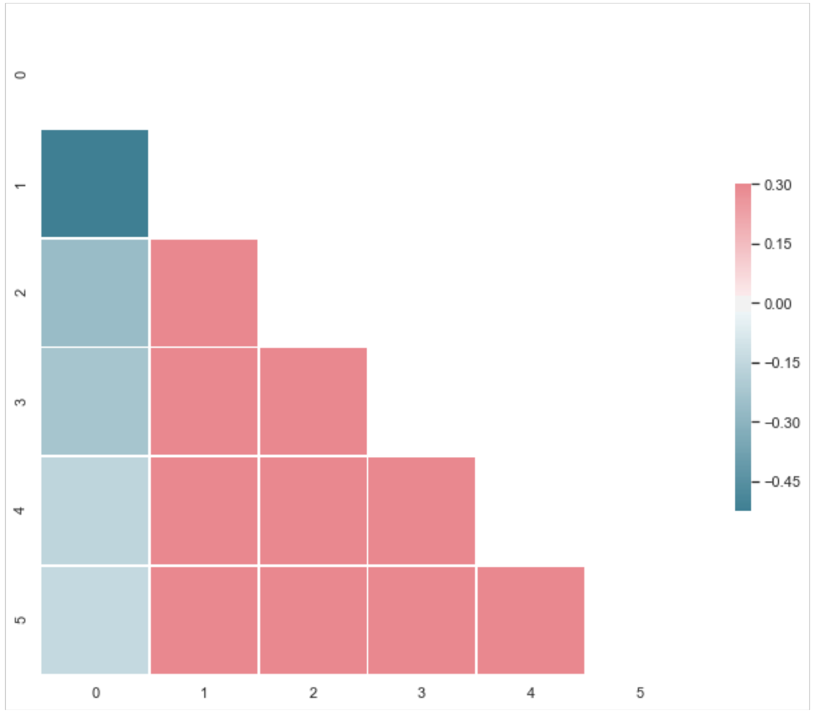

count
0
2000
4000
6000
8000
2.0 3.0 4.0 5.0 6.0 7.0 8.0 9.0 10.0
minor


0
2000
4000
6000
8000
10000
12000
0
1

EDA(Audio features)

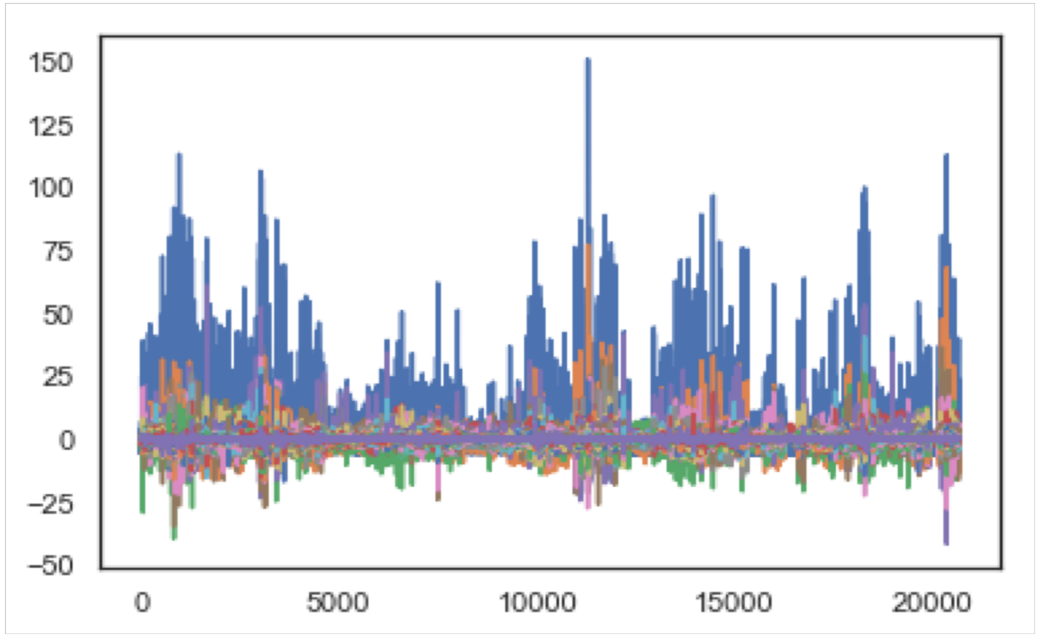
- **154 features**

  PCA or not?



University of Pittsburgh

EDA(Audio features)

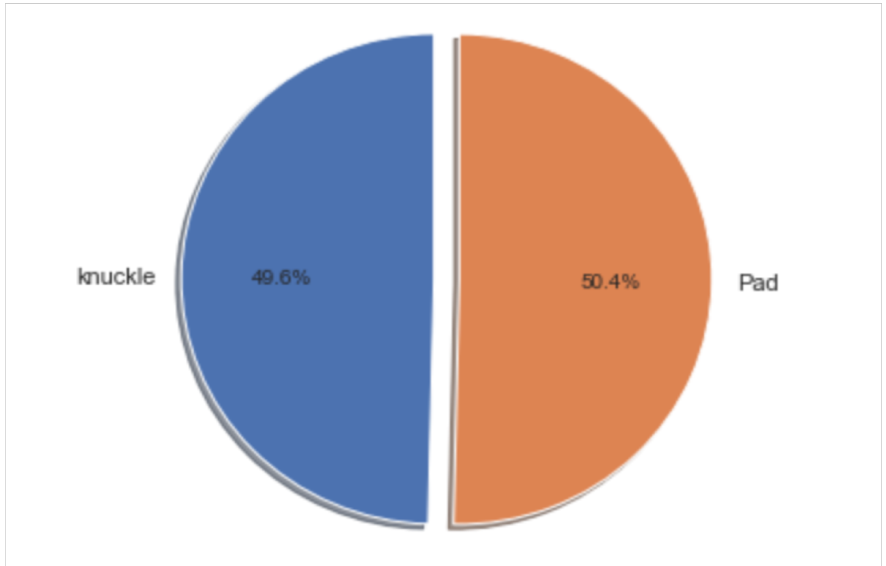


❖ Apply PCA?

- Audio Features reduced to **35,** explained variances **0.95**
- However, after compare models with and without PCA, we found out the final model would be a little bit better without PCA

Model(prepare)

- ❖ **Check labels balance**



- ❖ **Split data**
  - Training set
  - Testing set
- ❖ **Scale(normalize) data**
  - reduce outliers
  - Avoid dominated by some features

Model Result

- **Apply Random Forest**
  - Out of bag score : 0.9501
  - RandomForestClassifier(n_estimators = 50, max_features='auto', criterion = 'entropy')

Model Result

- **Apply SVM**
  - Cross validation results
  - Accuracy: 0.95 (+/- 0.00)
  - Test set result: 0.9523
  - SVC(kernel = 'rbf', C = 10, gamma = 0.01)

```
cross validation score
[0.95251059 0.95190563 0.94765507 0.94856278 0.94916793]
cross validation accuracy: 0.9500 (+/- 0.00)

AUC Score (Test): 0.952348
Testset report
             precision    recall  f1-score   support

          0       0.96      0.95      0.95      2112
          1       0.95      0.95      0.95      2020

avg / total       0.95      0.95      0.95      4132

Testset confusion matrix
[[2009  103]
 [  94 1926]]
Testset accuracy
0.952323
```

Model Result

- **Apply Xgboost(The best)**
  - Cross validation results
  - Accuracy: 0.96 (+/- 0.00)
  - Test set result: 0.9699
  - XGBClassifier(booster = 'gbtree', learning_rate = 0.3, max_depth = 6)

```
cross validation score
[0.96249244 0.9646098  0.96187595 0.96187595 0.96248109]
cross validation accuracy: 0.9627 (+/- 0.00)

AUC Score (Test): 0.969975
Testset report
             precision    recall  f1-score   support

          0       0.97      0.97      0.97      2112
          1       0.97      0.97      0.97      2020

avg / total       0.97      0.97      0.97      4132

Testset confusion matrix
[[2050   62]
 [  62 1958]]
Testset accuracy
0.96999
```

Evaluation(cross validation)

As we chose the **Xgboost model** as the final model, we need to better evaluate again.
We can either choose test set(hold out during training) performance or cross validation.
If we use cross validation, we can make full use of all data, here we did a stratified cross validation to estimate the model, to make the performance more stable(robust), we want to try a bigger K for cross validation. Due to computation and time cost, we set k = 30 for this cross validation.

```
cross validation score
[0.87082729 0.97677794 0.99419448 0.92162554 0.88243832 0.94629898
 0.93033382 0.95645864 0.97387518 0.99709724 0.99274311 0.98548621
 0.97822932 0.99274311 0.9245283  0.97532656 0.98693759 0.97822932
 0.95500726 0.94629898 0.93904209 0.93323657 0.90566038 0.89114659
 0.93313953 0.96215429 0.92285298 0.98981077 0.99126638 0.91848617]
cross validation accuracy: 0.9517 (+/- 0.07)
```

As we can see, the performance did decrease a little bit, but still over 0.95. I would say it should be a relatively powerful model.

Some Thoughts

In the feature extraction part, After FFT, we have some features which are complex numbers. If directly use it as model input, the model will discard the imagine part which obviously would lose some information.
I tried to regard one complex number feature as two features(real part and imaginary part) instead, the model performance is bad.
I also tried to use absolute value(length) and phase as the features. The performance is good.
Then I transferred the complex number into its absolute value(length). The model showed a better results.
However, I believe there could exist a better way to fix this problem. By doing this, we should have a deep domain knowledge about the acoustic and complex number input for SVM.

Thank you!

QUESTIONS?