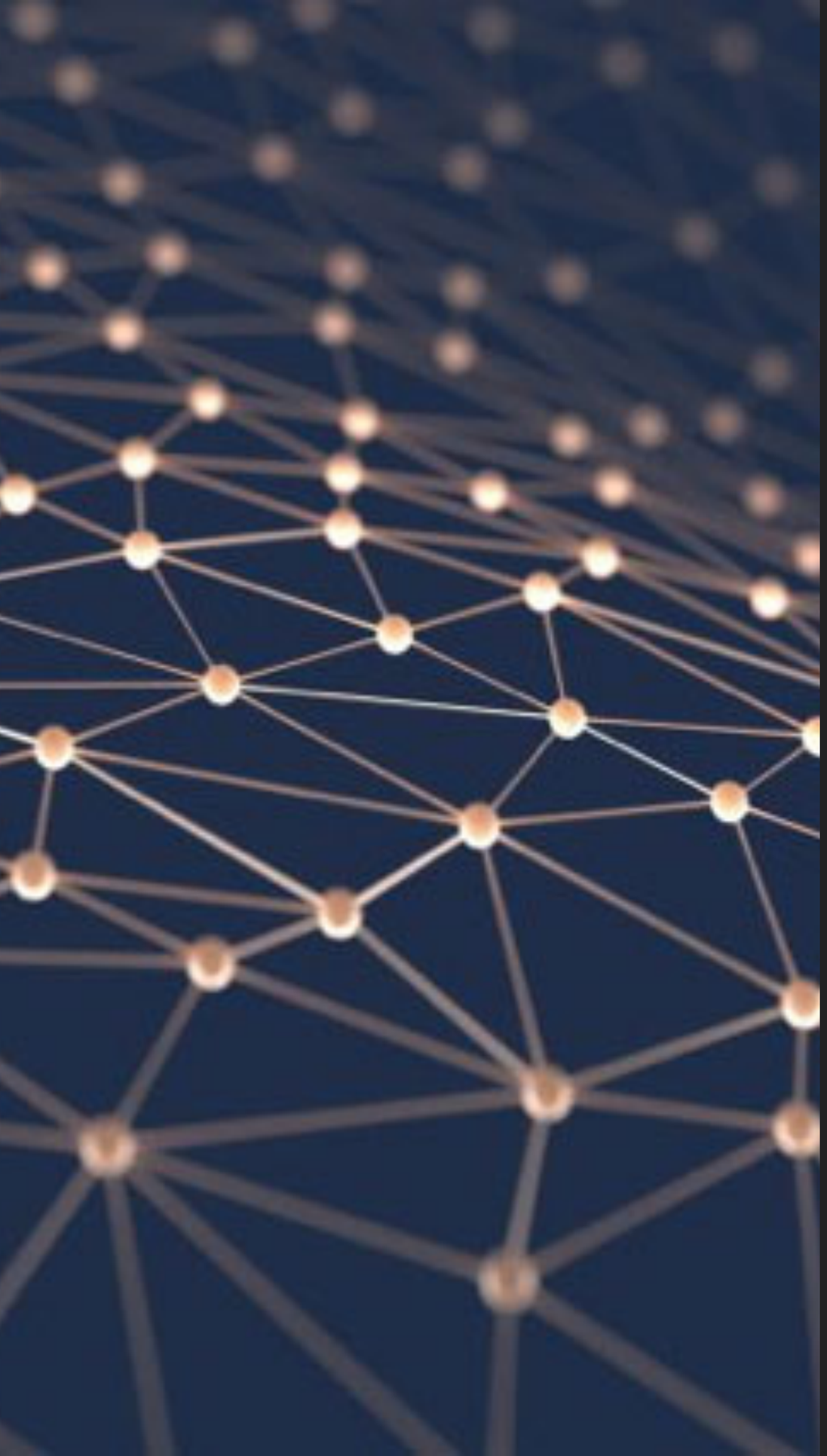
Αποστόλης Φανάκης 8261
Φρανκ Μπλάννινγκ 6698
Χριστίνα Θεοδωρίδου 8055

ΤΕΧΝΟΛΟΓΙΑ ΗΧΟΥ ΚΑΙ ΕΙΚΟΝΑΣ

OUR SPEECH / MUSIC CLASSIFIER



OUR PROCESS

WHAT DID WE DO?

GTZAN DATA SET

- 120 tracks
- 30 sec long
- 22050Hz
- Mono 16 bit
- .au and .wav format

http://opihi.cs.uvic.ca/sound/music_speech.tar.gz

```
$ ls music
bagpipe.au     deedee1.au     mingus1.au
ballad.au      deedee.au      mingus.au
bartok.au      duke.au        misirlou.au
beat.au        echoes.au      moanin.au
beatles.au     eguitar.au     narch.au
bigband.au     georose.au     ncherry.au
birdland.au    gismonti.au    nearhou.au
blues.au       glass1.au      opera1.au
bmarsalis.au   glass.au       opera.au
brahms.au      gravity2.au    pop.au
canonaki.au    gravity.au     prodigy.au
caravan.au     guitar.au      redhot.au
chaka.au       hendrix.au     rock2.au
classical1.au  ipanema.au     rock.au
classical2.au  jazz1.au       russo.au
classical.au   jazz.au        tony.au
copland2.au    led.au         u2.au
copland.au     loreena.au     unpoco.au
corea1.au      madradeus.au   vlobos.au
corea.au       magkas.au      winds.au
cure.au        march.au
debussy.au     marlene.au
```

```
$ ls speech
acomic2.au     fire.au        pulp1.au
acomic.au      geography1.au  pulp2.au
allison.au     geography.au   pulp.au
amal.au        georg.au       relation.au
austria.au     god.au         serbian.au
bathroom1.au   greek1.au      shannon.au
chant.au       greek.au       sleep.au
charles.au     india.au       smoke1.au
china.au       jony.au        smoking.au
comedy1.au     jvoice.au      stupid.au
comedy.au      kedar.au       teachers1.au
conversion.au  kid.au         teachers2.au
danie1.au      lena.au        teachers.au
danie.au       male.au        thlui.au
dialogue1.au   my_voice.au    undergrad.au
dialogue2.au   nether.au      vegetables1.au
dialogue.au    news1.au       vegetables2.au
diamond.au     news2.au       vegetables.au
ellhnika.au    nj105a.au      voice.au
emil.au        nj105.au       voices.au
female.au      oneday.au
fem_rock.au    psychic.au
$
```

OUR STACK

- Python
  - **Essentia** *for feature extraction*
  - **scikit-learn** *for preprocessing and classification*
  - **Seaborn** *for visualization*
- GitHub for collaboration

```
$ tree -L 2 .
.
├── classification_model_training
│   └── __pycache__
├── compined.wav
├── feature_extraction
│   ├── batch_feature_extractor.py
│   ├── feature_extractor.py
│   ├── __init__.py
│   ├── music_features
│   ├── __pycache__
│   ├── README.md
│   └── speech_features
├── featuresStream
│   └── tmp.json
├── pipeline.py
├── preprocessing
│   ├── data_preprocessing.py
│   ├── dataset.pkl
│   ├── __init__.py
│   ├── __pycache__
│   └── README.md
├── test
│   ├── accuracySingleFeature.py
│   └── accuracyWithoutFeature.py
├── tmp.json
├── training
│   ├── __init__.py
│   ├── model_training.py
│   └── __pycache__
└── visualization
    ├── output
    └── visualization.py

14 directories, 17 files
$
```

FEATURE EXTRACTION

- **feature_extractor.py**
  - 6144 sample window
  - a set of 27 features
- **batch_feature_extractor.py**
  - For all the files!

```
$ ls music_features/
bagpipe.json     deedee1.json     mingus1.json
ballad.json      deedee.json      mingus.json
bartok.json      duke.json        misirlou.json
beat.json        echoes.json      moanin.json
beatles.json     eguitar.json     narch.json
bigband.json     georose.json     ncherry.json
birdland.json    gismonti.json    nearhou.json
blues.json       glass1.json      opera1.json
bmarsalis.json   glass.json       opera.json
brahms.json      gravity2.json    pop.json
canonaki.json    gravity.json     prodigy.json
caravan.json     guitar.json      redhot.json
chaka.json       hendrix.json     rock2.json
classical1.json  ipanema.json     rock.json
classical2.json  jazz1.json       russo.json
classical.json   jazz.json        tony.json
copland2.json    led.json         u2.json
copland.json     loreena.json     unpoco.json
corea1.json      madradeus.json   vlobos.json
corea.json       magkas.json      winds.json
cure.json        march.json
debussy.json     marlene.json
$ ls speech_features/
acomic2.json     fire.json        pulp1.json
acomic.json      geography1.json  pulp2.json
allison.json     geography.json   pulp.json
amal.json        georg.json       relation.json
austria.json     god.json         serbian.json
bathroom1.json   greek1.json      shannon.json
chant.json       greek.json       sleep.json
charles.json     india.json       smoke1.json
china.json       jony.json        smoking.json
comedy1.json     jvoice.json      stupid.json
comedy.json      kedar.json       teachers1.json
conversion.json  kid.json         teachers2.json
danie1.json      lena.json        teachers.json
danie.json       male.json        thlui.json
dialogue1.json   my_voice.json    undergrad.json
dialogue2.json   nether.json      vegetables1.json
dialogue.json    news1.json       vegetables2.json
diamond.json     news2.json       vegetables.json
ellhnika.json    nj105a.json      voice.json
emil.json        nj105.json       voices.json
female.json      oneday.json
fem_rock.json    psychic.json
$
```

FEATURES (TIME DOMAIN)

- Zero Crossing Rate
- Log Attack Time
- Decay
- Flatness

FEATURES (CEPSTRAL)

- Mel Band Energies
- MFCC coefficients
- 4Hz Energy Modulation

FEATURES (SPECTRAL)

- Roll Off
- Spectral Flux
- Flatness SFX
- Flatness DB
- Attack time
- Decay
- High Frequency Content
- Spectral Complexity

OUR PREPROCESSING

- **data_preprocessing.py**
  - reads .json files
  - standardizes features' levels
  - applies PCA if requested
  - creates .pkl file with features of all samples for plotting

TRAINING

- SVM
- Decision Tree
- MultiLayer Perceptron
- Naive Bayes
- Random Forest

Stats will come later

WRAPPER

pipeline.py

```python
import numpy as np
import pandas as pd
from feature_extraction.feature_extractor import extractFeatures
from feature_extraction.batch_feature_extractor import batchExtract
from preprocessing.data_preprocessing import arrayFromJSON, standardization, PCA
from training.model_training import simpleTrain, kFCrossValid

musicFeatures = batchExtract('../dataset/music_wav/', 'feature_extraction/music_features/', 22050)
musicFeatures = musicFeatures.assign(target=0)
speechFeatures = batchExtract('../dataset/speech_wav/', 'feature_extraction/speech_features/', 22050)
speechFeatures = speechFeatures.assign(target=1)

dataset = pd.concat([musicFeatures, speechFeatures])
target = dataset.pop('target').values

dataset = standardization(dataset)
# dataset = PCA(dataset)

print('Simple train accuracy achieved = ' + str(simpleTrain(dataset, target)))
kFCrossValid(dataset, target, model = 'svm')
clf = kFCrossValid(dataset, target, model = 'rndForest')

features = extractFeatures('compined.wav', 'tmp.json', 22050)
features = standardization(features)
audioClass = clf.predict(features)
print(audioClass)
```

```
-:---  pipeline.py    All L1    Git-master  (Python Fly Compiling ElDoc)
<nil> <mouse-1> is undefined
```

EXECUTION

```
[   INFO   ] MusicExtractorSVM: no classifier models were configured by default
feature_extractor loaded
batch_feature_extractor loaded
feature_preprocessing loaded
model_training loaded
Batch feature extraction finished successfully.
Batch feature extraction finished successfully.
Running standardization
Simple train accuracy achieved = 0.9479784612576435
[fold 0], score: 95.78
[fold 1], score: 95.44
[fold 2], score: 95.62
[fold 3], score: 95.13
[fold 4], score: 95.09
[fold 0], score: 95.51
[fold 1], score: 95.15
[fold 2], score: 95.29
[fold 3], score: 95.24
[fold 4], score: 94.89
Running standardization
[0 1 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 1 0 1 0 1 0 0 0 0 1 1 1 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0
 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 1 0 1 0 1 0 0 0 0 0 0 0 1 0 1 1 1 0 0 0 0 0 0 1 0 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1]
```

THE AFTERMATH

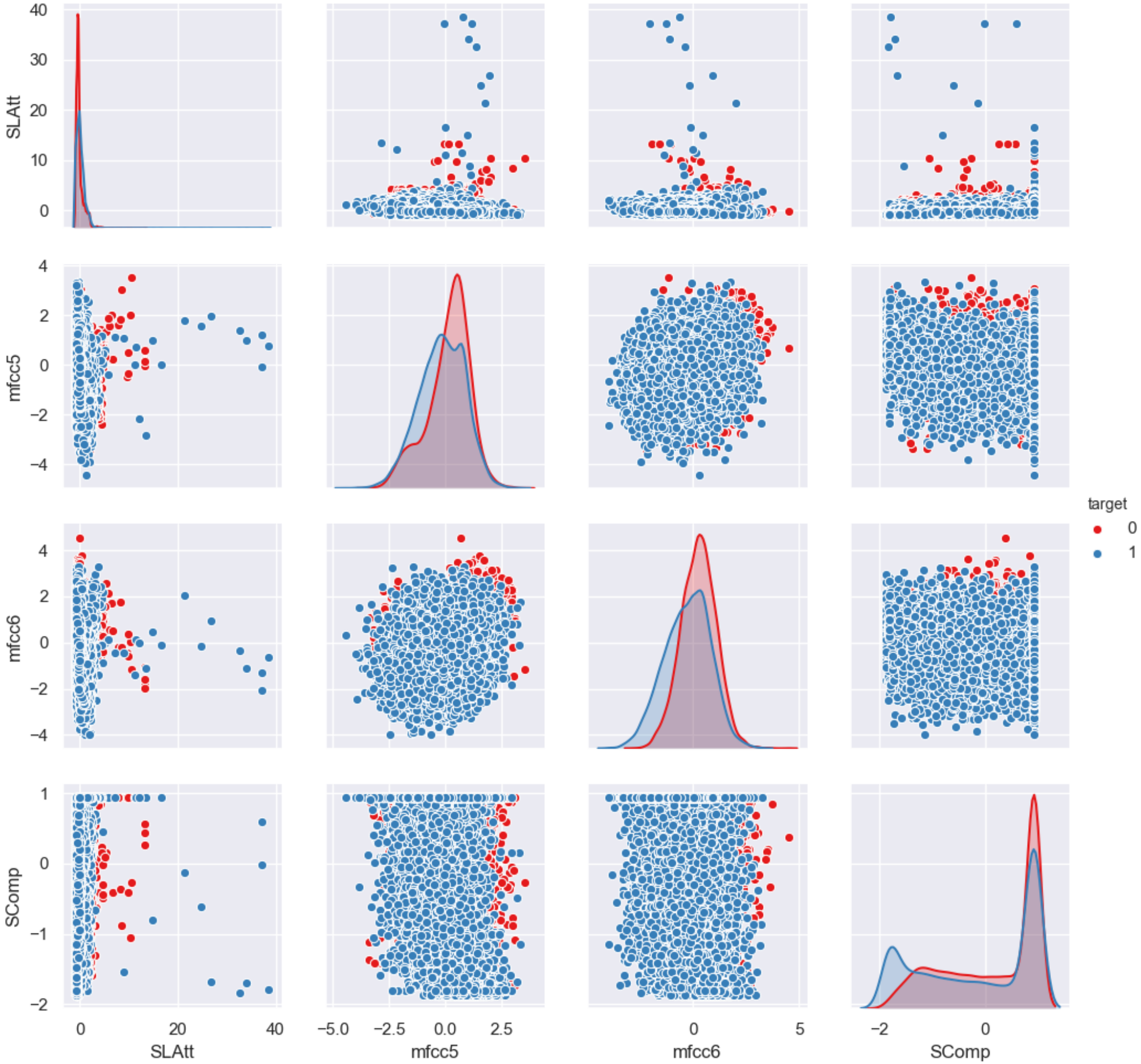
STATISTICS

TRAINING

Model	Accuracy
SVM	96.06
Decision Tree	85.96
MultiLayer Perceptron	90.34
Naive Bayes	70.25
Random Forest	95.49

TRAINING

Model	Accuracy
SVM	96.06
Decision Tree	85.96
MultiLayer Perceptron	90.34
Naive Bayes	70.25
Random Forest	95.49
SVM with PCA (10)	90.02


SLAtt
mfcc5
mfcc6
SComp
target
0
1

CONCLUSION

- The more features, the better
  - Most important features were:
    - SSDec 65% accu
    - MFCC 6 +7% accu
    - MFCC 3 +4% accu
- SVM and Random Forest models are the best
- With no time constraints, PCA doesn’ t make sence

THANK YOU

Αποστόλης, Φρανκ, Χριστίνα

https://github.com/laserscout/THE-Assignment