Emotion Detection

Group 1: Daniel Lee, Suzy Gao, Bingquan Wu, LuLu Dong, Tushar Ponkshe

Table of Content

Baseline Model - GBM

- Test set accuracy: 31%
- Train set accuracy: 62%
- Running time
    - Training: 8.42s
    - Predicting: 9.99s
- Limitation of GBM
    - GBMs are more sensitive to overfitting if the data is noisy.
    - Training generally takes longer because of the fact that trees are built sequentially.
    - GBMs are harder to tune than RF. There are typically three parameters: number of trees, depth of trees and learning rate, and each tree built is generally shallow.

Naive Bayes Classifier

- Test set accuracy: 22.4%
- Train set accuracy: 22.7%
- Running time
  - Training: 2.11s
  - Predicting: 6.78s
- Limitations:
  - The strong assumption about the features to be independent which is hardly true in real life applications.
  - Chances of loss of accuracy.
  - Zero Frequency i.e. if the category of any categorical variable is not seen in training data set then model assigns a zero probability to that category and then a prediction cannot be made.

XGboost

Best accuracy on test set: 33%

Best accuracy on train set: 55%

---------------- With PCA ---------------------

Best accuracy on test set: 34%

Best accuracy on train set: 47.45%

Running time:

training 18m 29s, predicting 6.75s

Applied 10-fold cross validation with Parameter Tuning using Grid Search

PCA does not improve XGboost model a lot as it's already a correlation robust algorithm!

*Both **xgboost** and **gbm** follows the principle of **gradient boosting**. There are however, the difference in modeling details. Specifically, xgboost used a **more regularized model formalization to control over-fitting**, which gives it better performance.*

Advanced Model: SVM

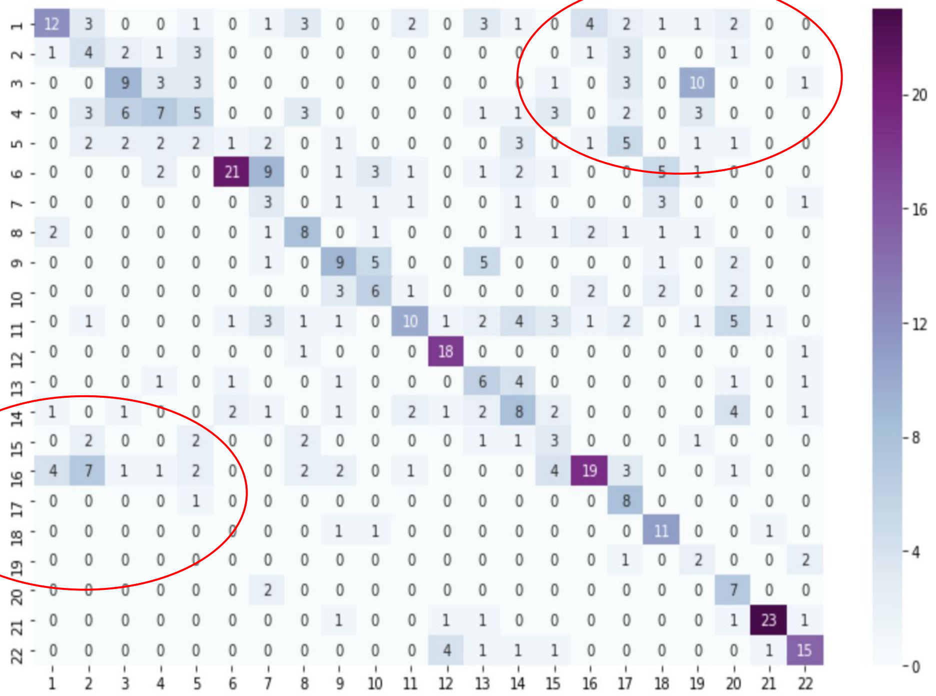
Accuracy on test set: 49%
Accuracy on train set: 99%

------------------- With PCA -------------------
- Reducing features by Keeping 94% of the original data.
- Using only 21 features selected by PCA

Accuracy on test set: 42.4%
Accuracy on train set: 47.2%
(applied 10-fold cross validation)

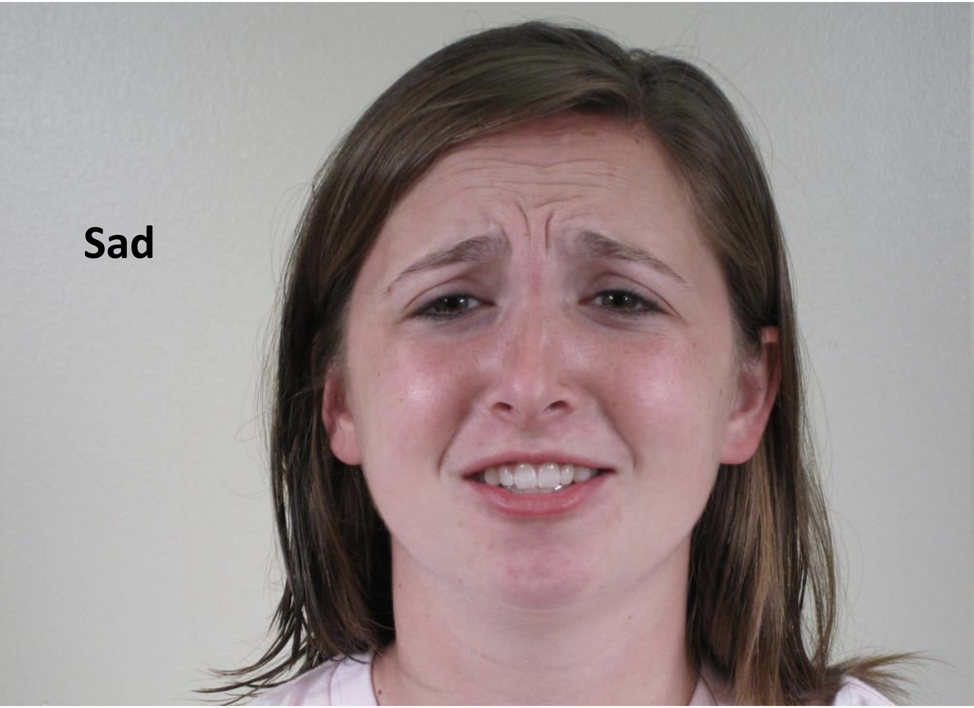
Confusion Matrix

Actual



Prediction

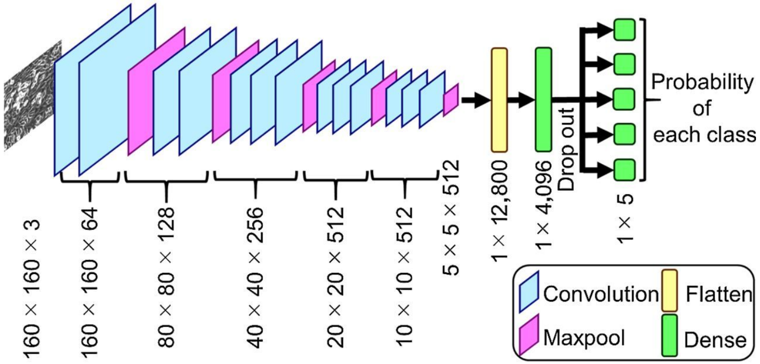
Take a guess?



Deep Learning - CNN

- Test set accuracy: 47.4%
- Train set accuracy: 47.3%
- Batch size: 200
- Epochs: 10
- Running time
    - Training: 17.06s
    - Predicting: 2.06s
- Limitations:
    - CNNs perform poorly with less data.
    - CNNs have millions of parameters and with small dataset, would run into an overfitting problem because they need massive amount of data to quench the thirst.

```
Model: "sequential_4"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv1d_10 (Conv1D)           (None, 35, 64)            256
_________________________________________________________________
conv1d_11 (Conv1D)           (None, 33, 64)            12352
_________________________________________________________________
conv1d_12 (Conv1D)           (None, 31, 64)            12352
_________________________________________________________________
flatten_4 (Flatten)          (None, 1984)              0
_________________________________________________________________
dense_7 (Dense)              (None, 100)               198500
_________________________________________________________________
dense_8 (Dense)              (None, 22)                2222
=================================================================
Total params: 225,682
Trainable params: 225,682
Non-trainable params: 0
_________________________________________________________________
```



Model Comparison & Conclusion

	GBM	XGB	Naive Bayes	SVM	CNN
Training Accuracy	62%	47.45%	22.9%	47.2%	47.3%
Test Accuracy	31%	34%	20.8%	42.4%	47.4%
Computational Time(train)	9.99s	18min 29s	6.78s	12.6s	17.06s
Computational Memory(train)	480 MiB	312.81 MiB	497 MiB	462.5 MiB	574.51MiB
Test running cost	8.42s	6.57s	4.4s	1.14s	2.06s