Facial Image classification

...

5243 Project3 - Group4
Yushi Pan, Yuqi Xing, Serena Yuan, Haosheng Ai, Chuyun Shu

Project Description

- Goal: Using machine learning models to make classification predictions about facial images: whether they have simple emotions or compound emotions.

- Criteria: We want to achieve as higher accuracy as possible, we also consider whether weighted, AUC, running time as the factors while choosing the model

Model Summary

- Baseline model: Gradient Boosting Machine
- Advanced model: XGBoost with SMOTE

- Other models we've tried:
- PCA + LDA
- Random Forest (unweighted)
- Random Forest (weighted)
- SVM (kernel = polynomial, linear, radial, sigmoid)
- KNN

Baseline: Gradient Boosting Machine

- Cross-validation on parameters:

  Shrinkage = 0.01, 0.05, 0.1

  N.trees = 500, 1000, 1500

- Best hyperparameter: shrinkage=0.05, n.trees = 1000
- Accuracy: 69.14%
- AUC: 0.80
- Time for training model: 206.574s
- Time for testing model: 14.67s

XGBoost (eXtreme Gradient Boosting) with SMOTE

Since we are dealing with the imbalanced data, there are too few examples of the minority class for a model to efficiently learn the decision boundary. We used SMOTE (Synthetic Minority Oversampling Technique) to generate the synthetic samples for the minority class.

XGBoost with SMOTE

Main hyperparameters include but not limited to:

eta (learning_rate) = 0.08

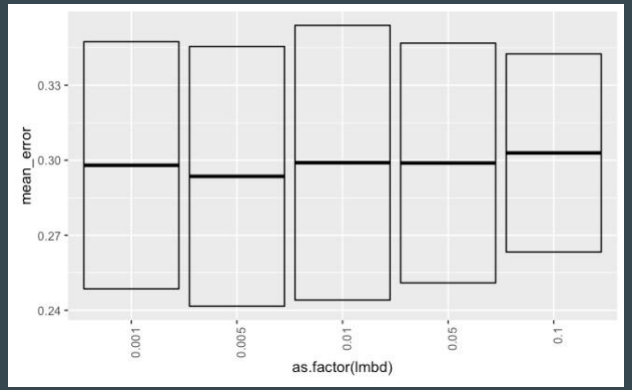
gamma (min_split_loss) = 0.4

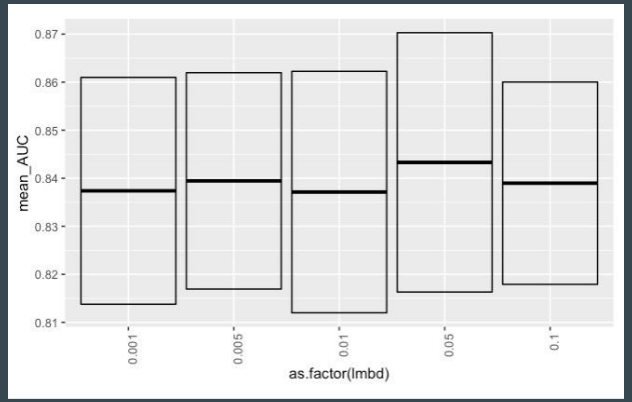
Max_depth = 5

lambda (reg_lambda) = 0.005

*nrounds (number of boosting iterations) = 200

*larger nrounds increase running time.





Result of XGBoost model

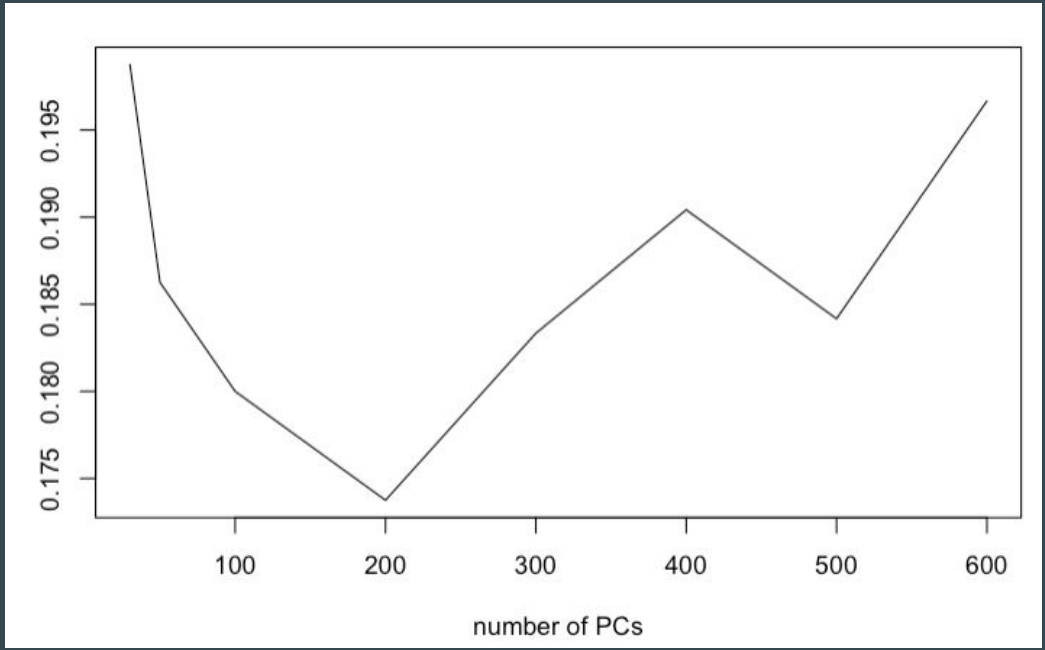
```
The accuracy of model: xgboost with lambda = 0.05 is 79.83333 %.
The AUC of model: xgboost with lambda = 0.05 is 0.7918771 .
Time for constructing training features= 1.299 s
Time for constructing testing features= 0.265 s
Time for training model= 168.574 s
Time for testing model= 0.207 s
```

Advantage: shorter training time, higher accuracy and AUC.

Too many tuning hyperparameters that hard to find the optimization among all hyperparameters' combinations.

PCA+LDA

- Cross-validation on number of principal components



Best number of pcs: 200

PCA+LDA

- Accuracy: 73%
- AUC: 0.524
- Time for training model: 0.503s
- Time for testing model: 0.015s

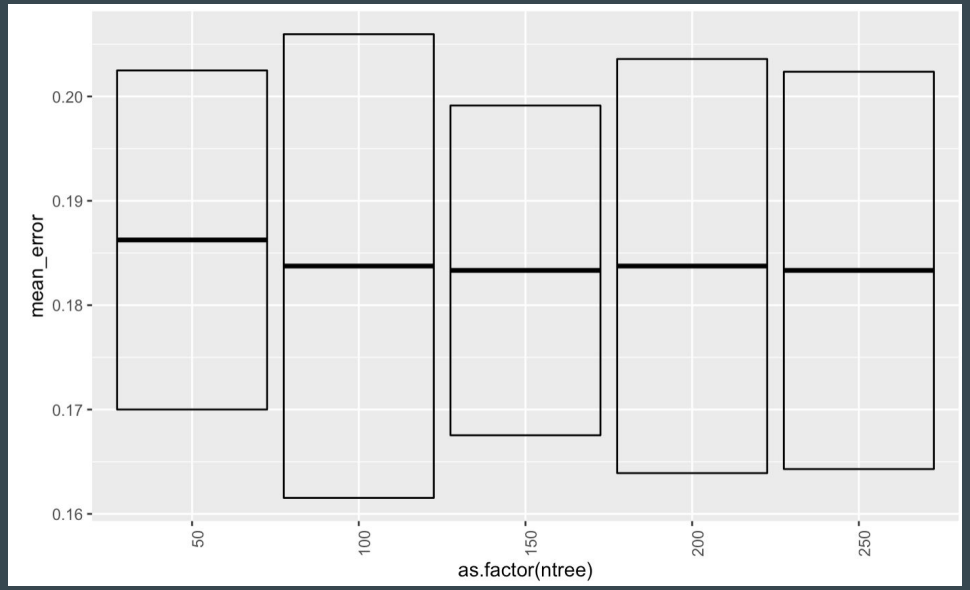
- Although the accuracy for the LDA model is higher than the baseline model and it trained the model very fast, the model is unweighted so it might be not proper to deal with the imbalanced data.

Random Forest

- Cross-validated ntrees:

ntrees: 50, 100, 150, 200, 250

- Accuracy: 79.3%
- Unweighted AUC: 0.54
- Train time: 176.072 s
- Test time: 0.166 s



Reason why Random Forest is not our advanced model:

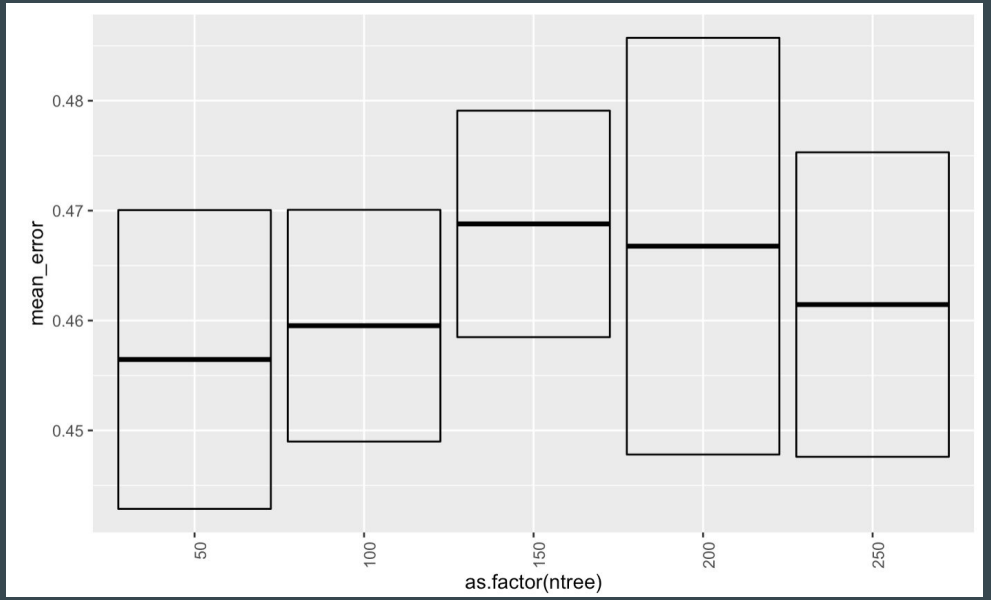
Unweighted model training imbalance data

Random Forest with weight

- Cross-validated ntrees:

ntrees: 50, 100, 150, 200, 250

- Accuracy: 54.2%
- Unweighted AUC: 0.54
- Train time: 60.28 s
- Test time: 0.12 s

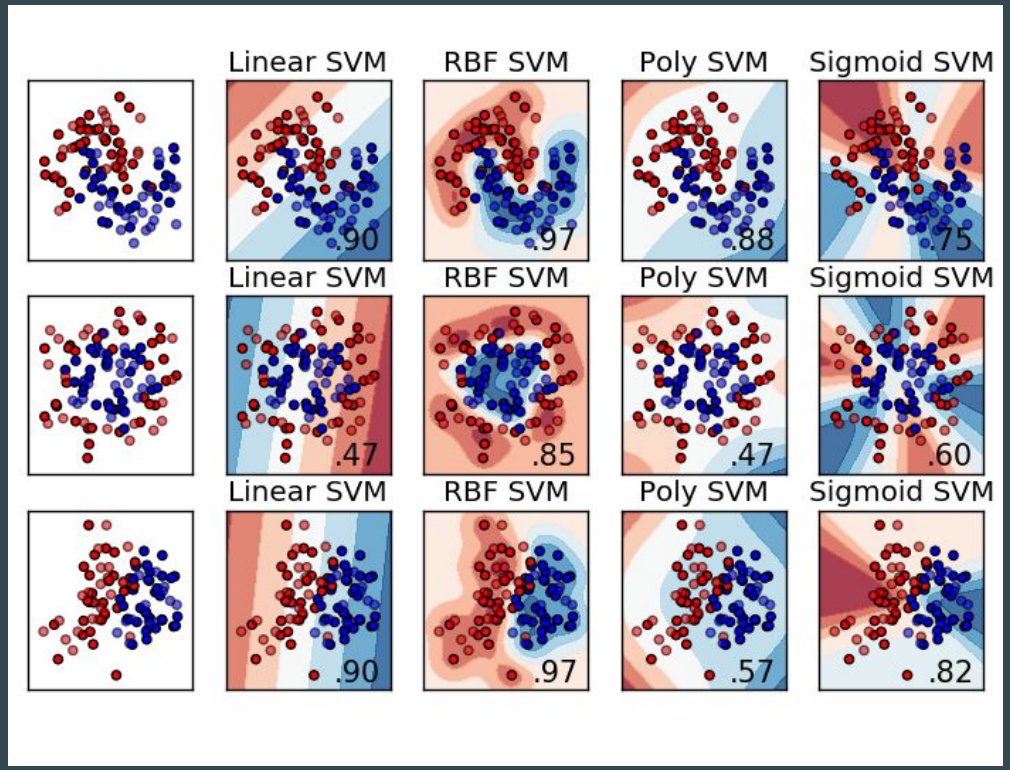


Reason why Random Forest with weight is not our advanced model:

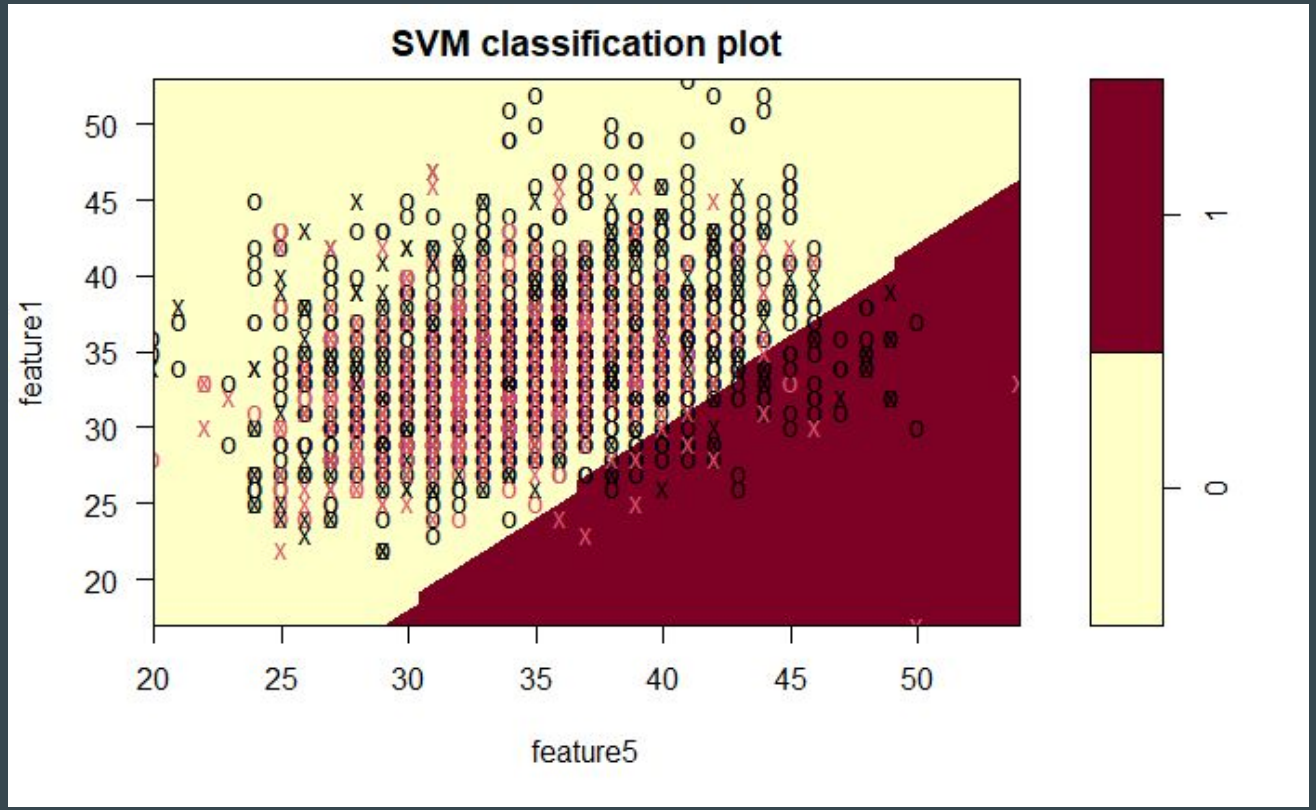
Accuracy and AUC not as good as XGBoost model

SVM: Linear, radial, polynomial, sigmoid

How the decision boundaries look depending on the kernel:



Linear SVM plot



Accuracy and auc of SVM models

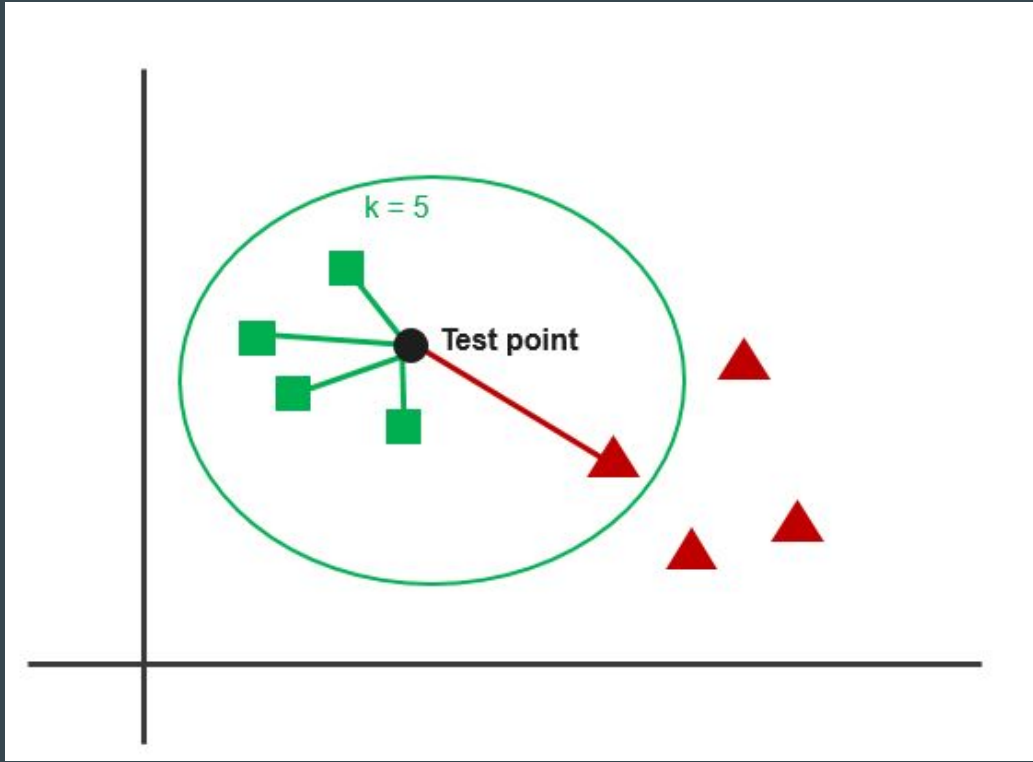
Non-probabilistic:

		Accuracy:
1.	Linear	94.83
2.	Polynomial	84.5
3.	Radial	82.66
4.	Sigmoid	74.33

Probabilistic:		Accuracy:	AUC:
1.	Linear	81	93
2.	Polynomial	82.66	92.18
3.	Radial	89.83	92.18
4.	Sigmoid	82.16	92.18

If the classification model (classifier) **is probabilistic**, **for** a given input, it will output **probabilities for** each class (**of** the N=2 classes) as the output. If the model **is Non-Probabilistic** (Deterministic), it will output only the most likely class that the input data instance belongs to.

K-Nearest Neighbor Model



Accuracy and auc of KNN model

	precision	recall	f1-score	support
0	0.48	0.52	0.50	25
1	0.65	0.61	0.63	36
accuracy			0.57	61
macro avg	0.56	0.57	0.56	61
weighted avg	0.58	0.57	0.58	61

From the classification report, it can be seen that the model has an average performance of around 57% ranging from precision, recall, f1-score, and support. Accuracy also shows in value of 57%.

Then for the AUC score, it can be seen that the value is around 56.5%.

Conclusion

Xgboost model with SMOTE is the one we chose as our final model because it yields higher accuracy and AUC than both the baseline model and the other models and it deals with the imbalanced data well.

Questions?