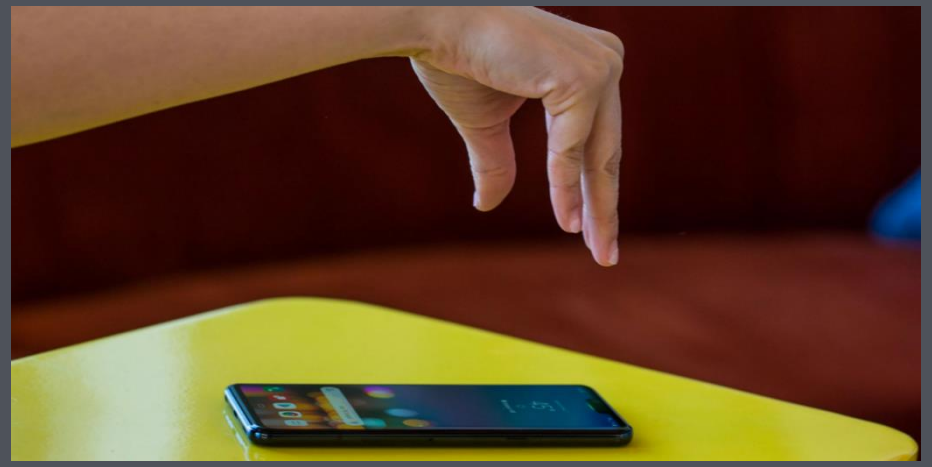
Hand Gesture Recognition

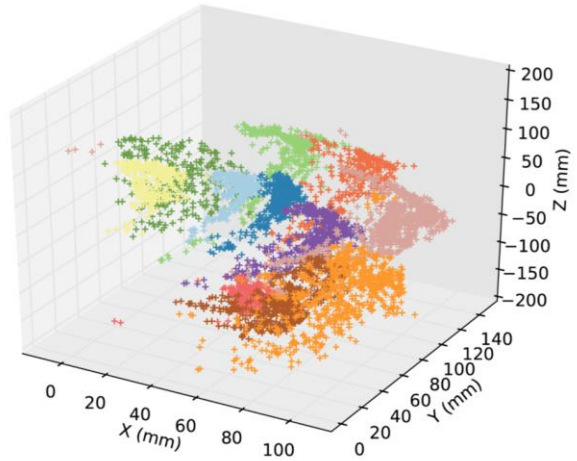
Cangcheng Tang
Brown University
12/4/2019
GitHub Link

Intro

- The purpose of this project is to build a model to predict hand gestures and users.
- Classification methods were used.
- Hand gesture recognition can be a very powerful tool in interaction with electronic devices.
- Obtained from UCI Machine Learning Repository.



Dataset Recap



Class of Gestures:

$1 \mapsto$ Fist (with thumb out),

$2 \mapsto$ Stop (hand flat),

$3 \mapsto$ Point1 (point with pointer finger),

$4 \mapsto$ Point2 (point with pointer and middle fingers),

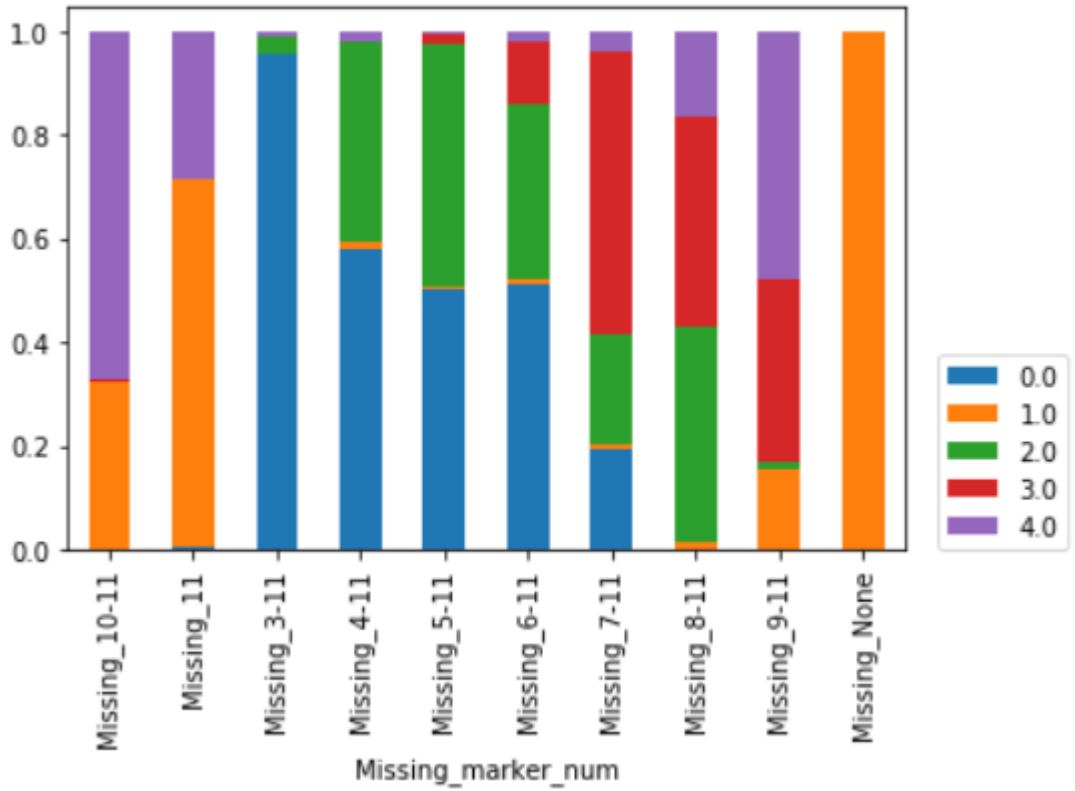
$5 \mapsto$ Grab (fingers curled as if to grab).

EDA Recap

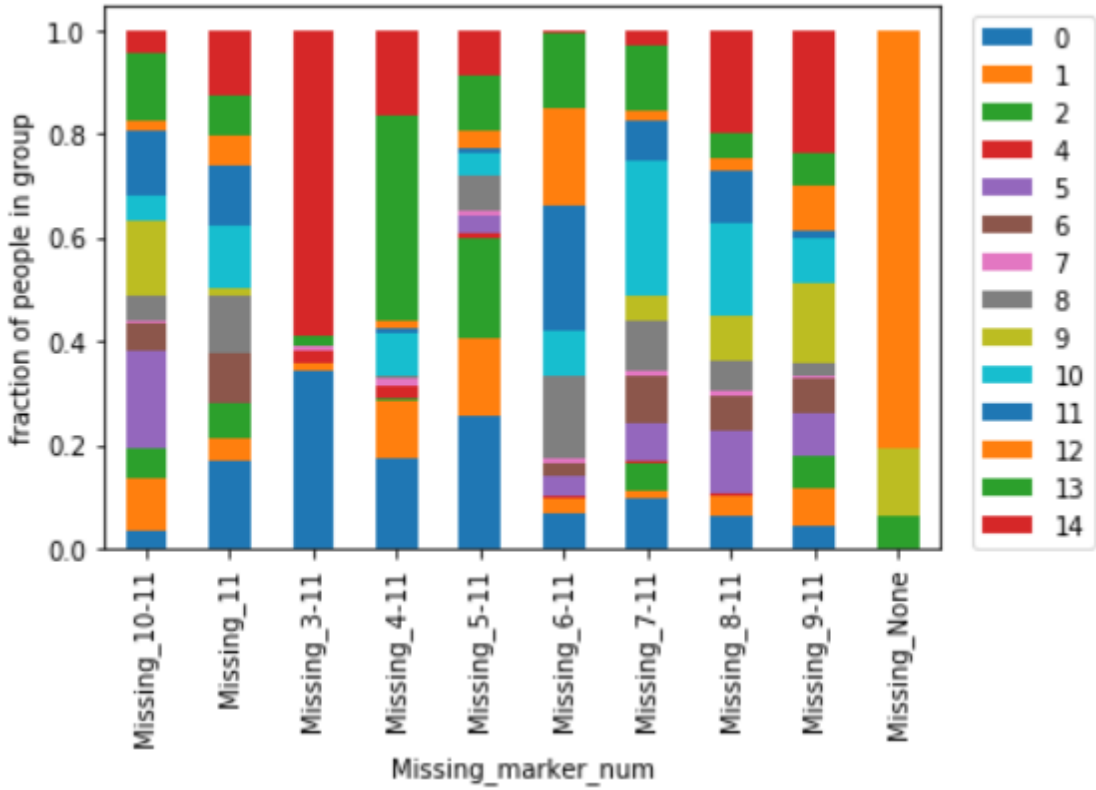
Missing values appear in most of the columns, from 5% to 100%, and are missing not at random.

To utilize this information, a new column is created to show which markers are missing at each data point.

Class of gestures



User



Cross Validation, Gesture Prediction

Split Data:

Data leakage can easily occur if a user appears in both train and cv or test.

To prevent that, train, cv and test sets were split by users. Train took up 60%, cv and test both took up 20%.

Preprocess:

- XGBoost: MinMaxScaler.
- SVC: MinMaxScaler, drop columns with more than 10% missing values, random forest to impute the rest.

Hyperparameters:

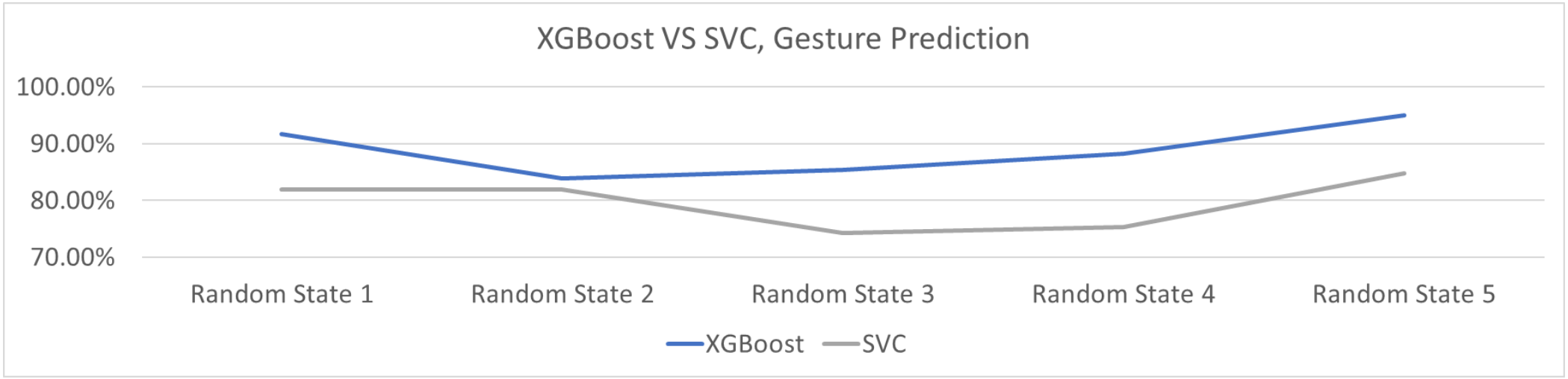
- XGBoost: λ and max_depth are tuned.
- SVC with rbf kernel: C and γ are tuned.

Results, Gesture Prediction

Accuracy:

XGBoost did better in both accuracy and standard deviation.

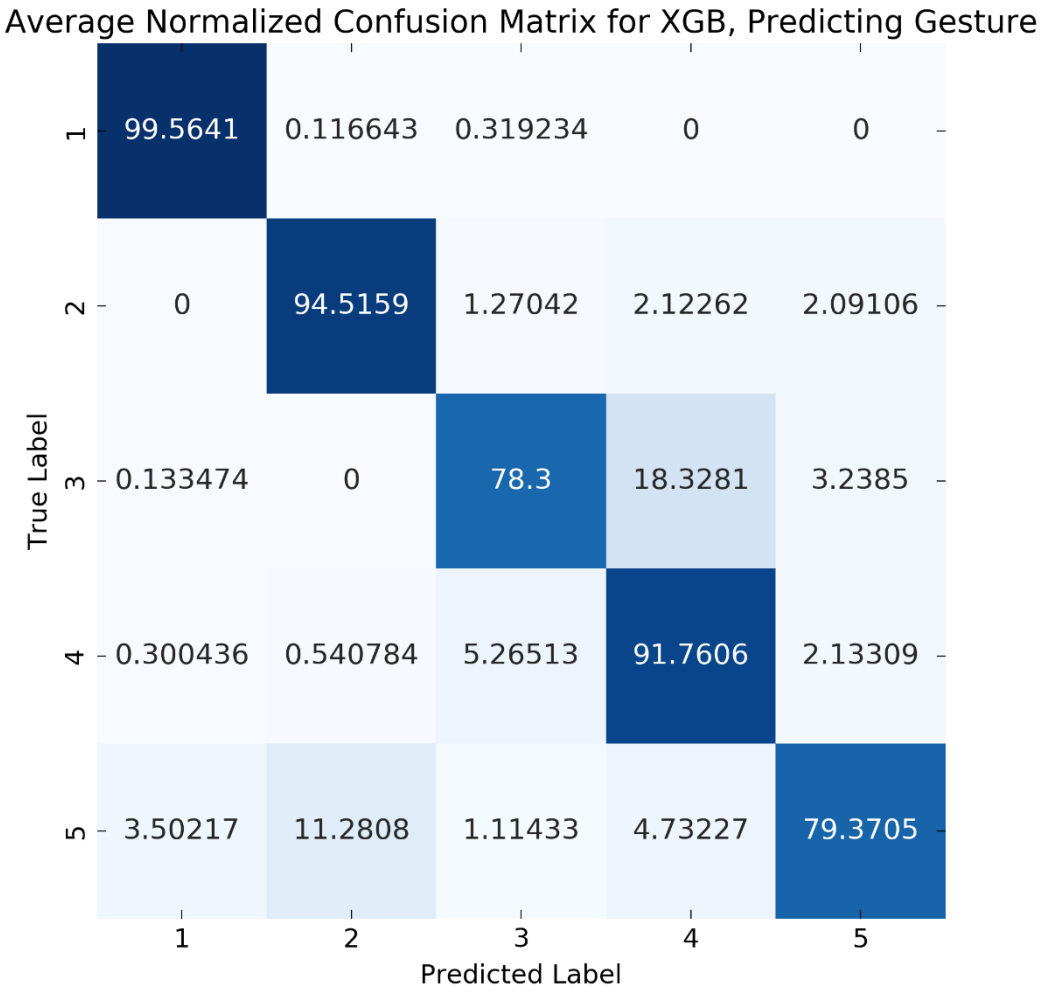
Model Comparison Summary, Predicting Gestures				
Model	**Baseline**	**Average accuracy**	**Standard deviation**	**Standard deviation above baseline**
XGBoost	20.93%	88.85%	0.0405	16.78
SVC		79.65%	0.0414	14.2

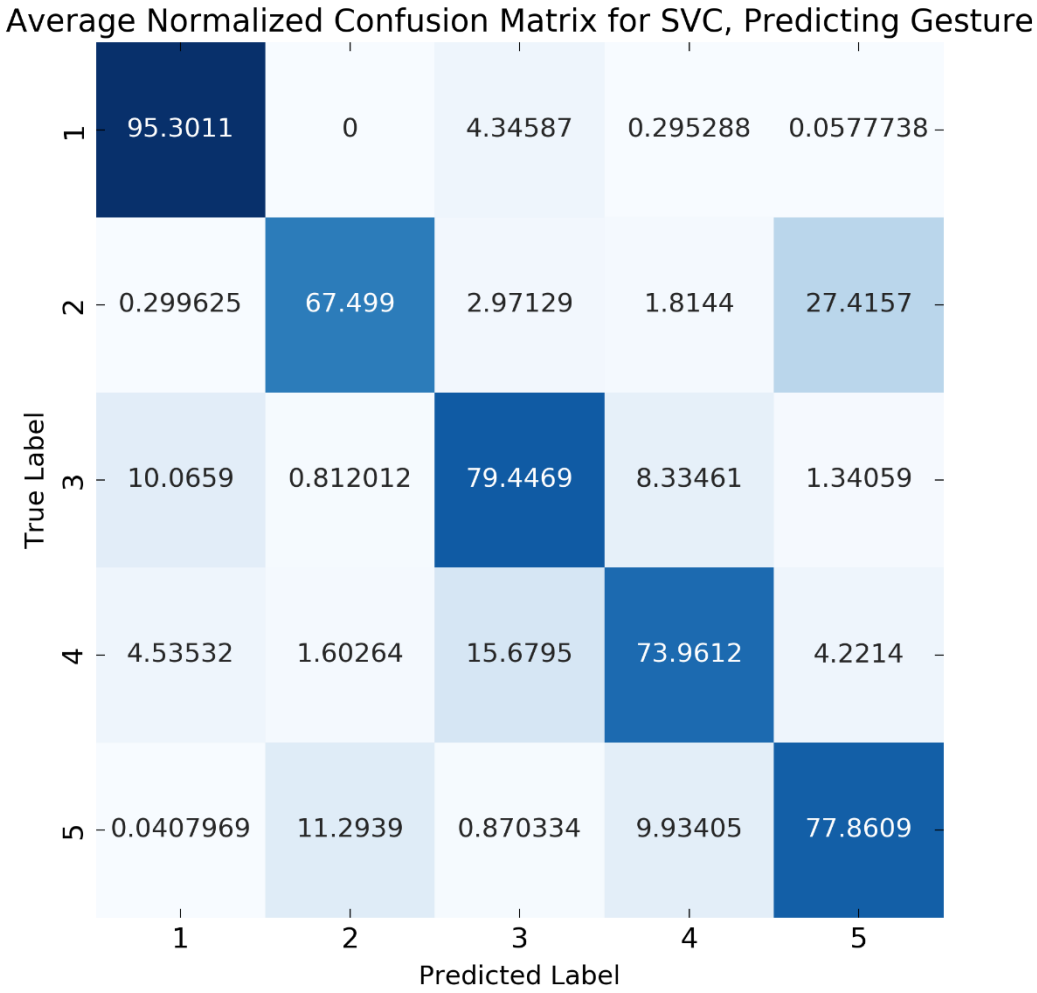


Results, Gesture Prediction

Confusion Matrix:

Confusions for XGBoost: (3, 4), (2, 5)
Confusions for SVC: (3, 4), (2, 5), (1, 3), (4, 5)





Results, Gesture Prediction

Local Feature importance for XGBoost:

Force plot for XGBoost, ordered from Class 1 to 5. Actual Class is 1.

Cross Validation, User Prediction

1st method :

Split by gestures. Tried to predict users with gestures the model haven't seen. Didn't work, only got 35% accuracy.

2nd method :

Tried to predict users given a certain gesture, it worked:

- Split data by gesture, then split into train, cv, test. Train models within each gesture.
- Compare results and find the best gesture and model.

Preprocess & Hyperparameters :

Models used are still XGBoost and SVC with rbf kernel.

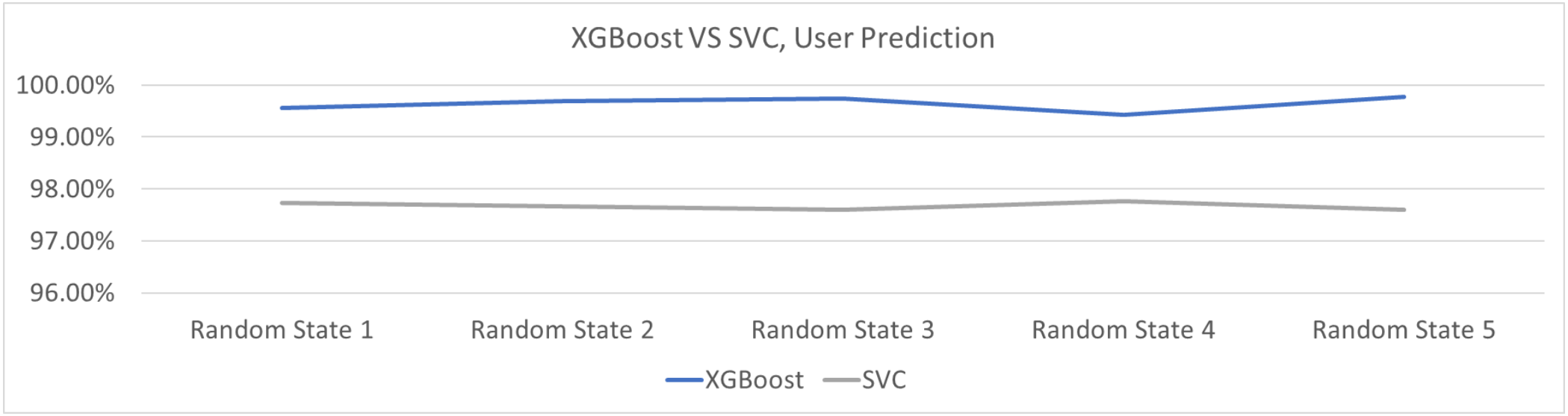
Basically the same as predicting gestures.

Results, User Prediction

Accuracy:

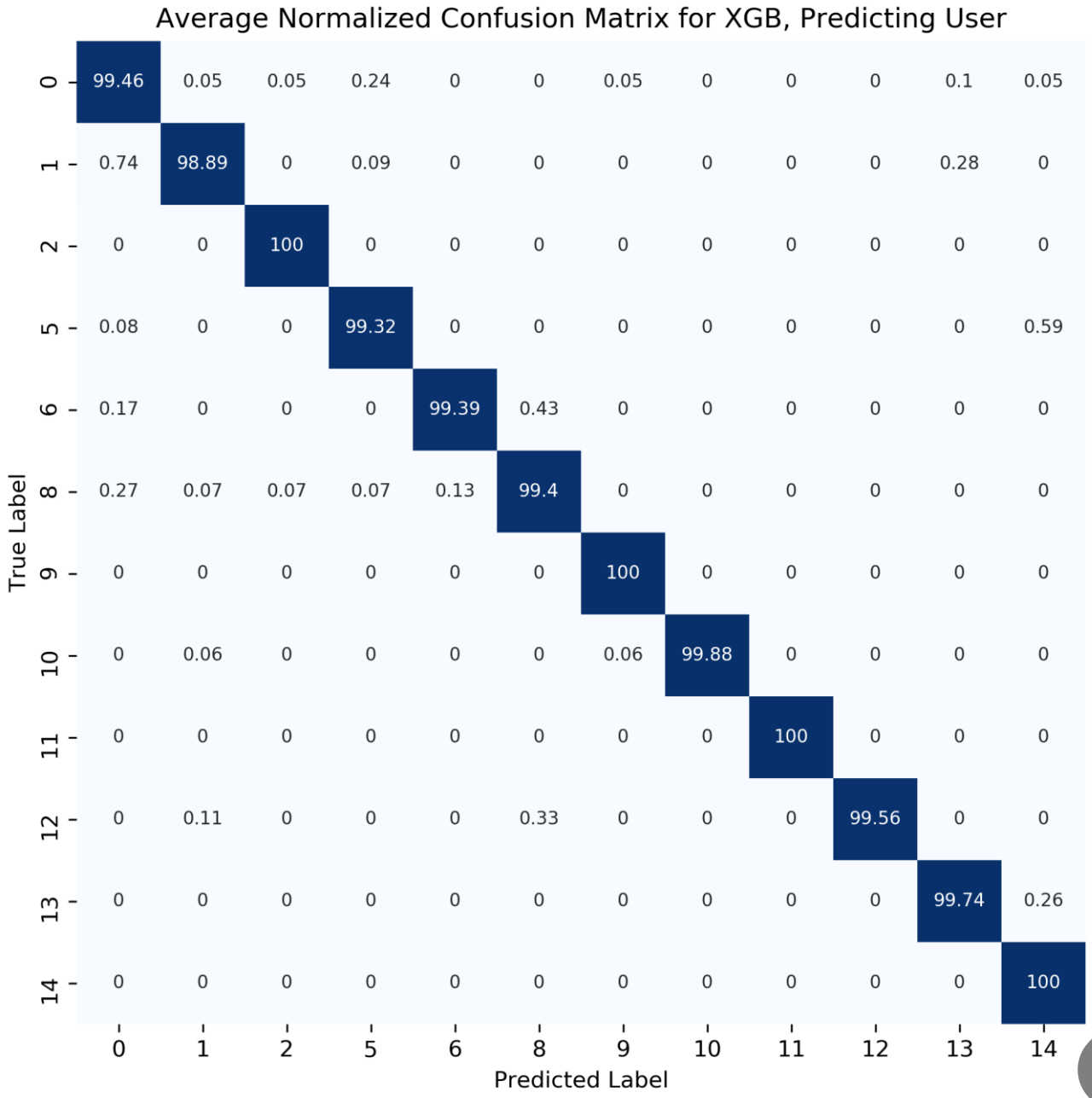
Both models did a good job. XGBoost did better in general.

Model Comparison Summary, Predicting Users					
Model	**Baseline**	**Best Gesture**	**Average accuracy**	**Standard deviation**	**Standard deviation above baseline**
XGBoost	12.26%	Class 5	99.64%	0.0013	674.11
SVC		Class 4	97.67%	0.0007	1237.48



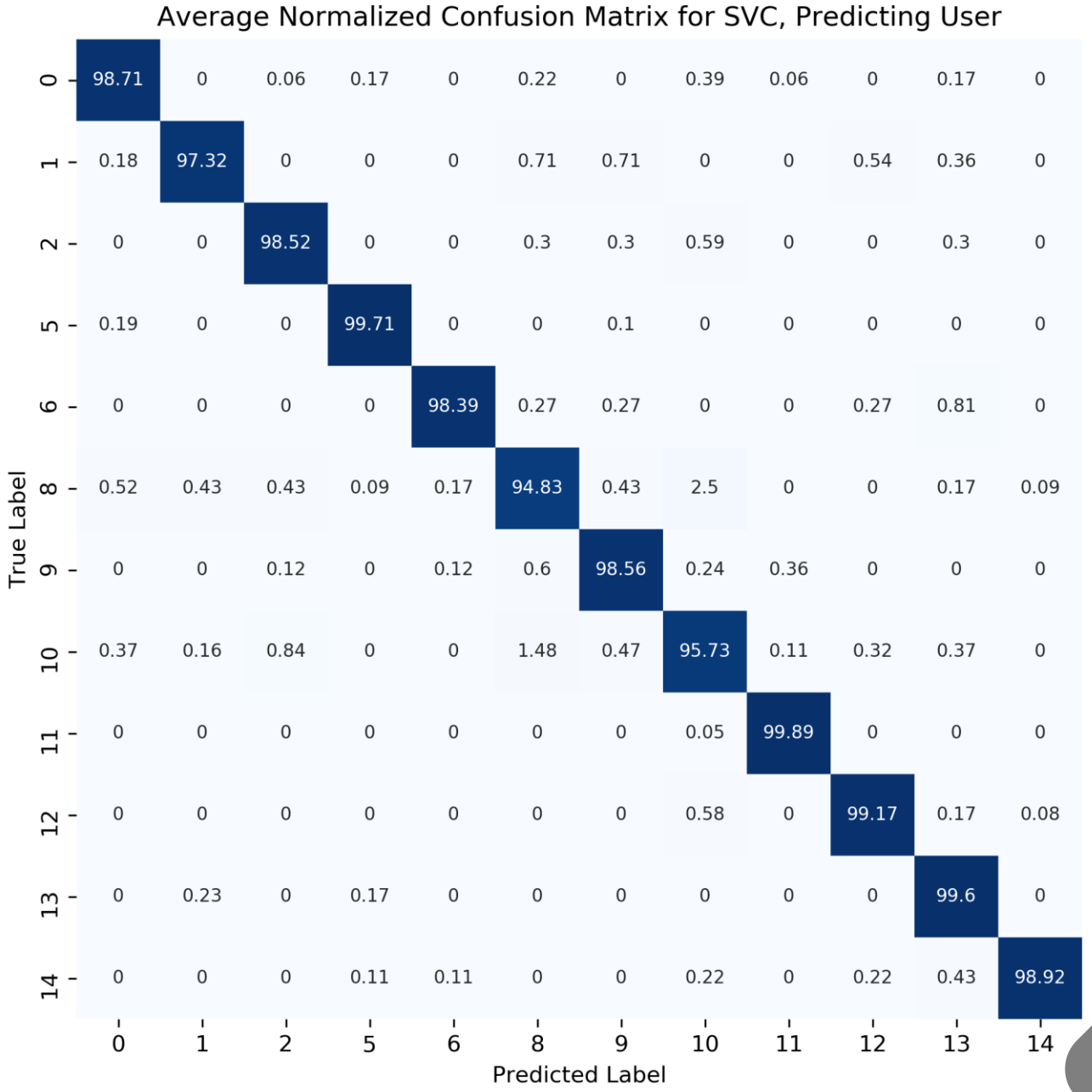
Results, User Prediction

Confusion Matrix for XGB:
Almost no confusions



Results, User Prediction

Confusion Matrix for SVC:
Almost no confusions



Results, User Prediction

Local Feature importance for XGBoost:

Force plot for XGBoost, the first plot is on User 0, the second plot is on User 1. Actual User is 1.

Outlook

For models:

- Tune hyperparameters more precisely.
- Try more machine learning models like neural network, logistic regression and KNN.

For features:

- Perform a more thorough EDA on coordinates and missing values.
- Try more feature engineering methods.


Q & A