1. **Data Preprocessing (using smaller dataset - 0.1 of original dataset)**

Type	F1_macro	Accuracy	Preicision_macro	Recall_macro
Pre1	0.656	0.676	0.713	0.676
Pre2	0.664	0.690	0.724	0.690
Pre3	0.661	0.687	0.719	0.687
Pre4	0.661	0.687	0.719	0.697
Pre5	0.661	0.687	0.719	0.687
Pre6	0.671	0.695	0.730	0.695

```
# pr1 = no data preprocessing
# pre2 = replace, remove_punctionation, lower
# pre3 = replace, remove_punctionation, lower, stopword
# pre4 = replace, remove_punctionation, lower, stopword, noisy
# pre5 = replace, remove_punctionation, lower, stopword, noisy,
frequency_words, scared_word
# pre6 = replace, remove_punctionation, lower, stopword, noisy,
frequency_words, scared_word, lemmatization
```

2. **Feature Engineering**

Preprocessing 6

Type	F1_macro	Accuracy	Preicision_macro	Recall_macro
TF-IDF	0.671	0.695	0.730	0.695
Word2Vec	0.531	0.560	0.565	0.559
LIWC	0.484	0.497	0.483	0.497
TF-IDF + LIWC	0.513	0.528	0.507	0.528

Besides TF-IDF + LIWC, other methods all use the smaller dataset (0.1 of the original dataset)

Maybe TF-IDF + LIWC exists the overfitting？

3. **Hyperparameters Tuning (using smaller dataset - 0.1 of original dataset)**

(1) C

C	0.0001	0.001	0.01	0.1	1.0	5.0	10.0
F1_macro	0.1	0.1	0.1	0.451	0.671	0.707	0.712

C	12.0	15.0	18.0	20.0	25.0	30	40	45
F1_macro	0.712	0.713	0.713	0.713	0.714	0.715	0.717	0.7173

C	60.0	70.0	80.0
F1_macro	0.7169	0.7174	0.7169

We can choose [40, 50,60,70] for 'C'

(2) Max_iter

Max_iter	100	500	1000	1500	2000	3000
F1_macro	0.716	0.7173	0.7173	0.7173	0.7173	0.7173

We can observe that max_iter does not influence much for the result.
We choose [500,1000,2000] for 'Max_iter'

(3) Solver

Solver	newton-cg	sag	saga	lbfgs
F1_macro	0.7173	0.7164	0.7165	0.7173

We choose ['newton-cg', 'lbfgs'] for 'Solver'

4. Final Training (using the original dataset)

Feature Engineering	C	max_iter	solver	F1	Accuracy	Precision	Recall
TF-IDF	40	1000	newton-cg	0.7493	0.758	0.775	0.758
TF-IDF	50	1500	newton-cg	0.7489	0.757	0.774	0.757

5. Original Dataset and Smaller dataset

Dataset	Feature Engineering	C	max_iter	solver	F1	Accuracy	Precision	Recall
Original	TF-IDF	40	1000	newton-cg	0.749	0.758	0.775	0.758
Smaller	TF-IDF	40	1000	newton-cg	0.7132	0.7207	0.7388	0.7207

Feature Engineering: TF-IDF + LIWC
Hyperparameter tuning:

(1) C

C	0.01	0.1	1.0	10.0	20.0	30.0	40.0
F1_macro	0.575	0.536	0.517	0.522	0.523	0.523	0.522

C	50	60	70	80
F1_macro	0.522	0.522	0.5226	0.5225

C=20, max_iter=1000, no normalization
[0.43392543555908913]

C=20, max_iter=1000, no normalization PCA
[0.41105864096998207]

TF-IDF

Shuffle	Feature Engineering	C	max_iter	solver	F1	Accuracy	Precision	Recall
	TF-IDF	40	1000	newton-cg	0.7493	0.758	0.775	0.758
Yes	TF-IDF	40	1000	newton-cg	0.969	0.971	0.973	0.966

(1) Training Dataset: All training data / Test Dataset: BalancedTest dataset
(2) Training Dataset: 80% Training dataset / Test Dataset: 20% Training dataset
(Training dataset is shuffled and them divided into two parts to get the result)

TF-IDF+LIWC

Data	Shuffle	C	max_iter	solver	F1	Accuracy	Precision	Recall
test		0.01	1000	newton-cg	0.5439	0.5467	0.5426	0.5467
train2	No	0.01	1000	newton-cg	0.026	0.0547	0.25	0.014
train2	Yes	0.01	1000	newton-cg	0.853	0.849	0.866	0.856
train1	Yes	0.01	1000	newton-cg	0.8773			

(1) Training dataset: All training data / Testing dataset: Balanced Test dataset
(2) Simply pick 45854 as training data and 3000 as test data from the training dataset
(3) Shuffle all training dataset -> Pick 45854 as training data and 3000 as test data from the training dataset
(4) Shuffle all training dataset -> Pick 45854 as training data and test data from the training dataset

TF-IDF

Train	Shuffle	C	max_iter	solver	F1	Accuracy	Precision	Recall
test		40	1000	newton-cg	0.7493	0.758	0.775	0.758
train2	Yes	40	1000	newton-cg	0.969	0.971	0.973	0.966
train1	Yes	40	1000	newton-cg	0.99994			

(1) Training Dataset: All training data / Test Dataset: BalancedTest dataset
(2) Training Dataset: 80% Training dataset / Test Dataset: 20% Training dataset
(Training dataset is shuffled and them divided into two parts to get the result)
(3) Training Dataset: 80% Training dataset / Test Dataset: 80% Training dataset
(Training dataset is shuffled and them divided into two parts to get the result)

XGBoost
Feature Engineering: TF-IDF

1. First Trial (10% of training dataset)
n_estimators = 100, learning_rate=1.0, max_depth=1, radom_sate=0

	F1	Accuracy	Precision	Recall
XGBoost	0.562	0.573	0.580	0.573

2. Hyperparameter Tuning

(1) Learning Rate

Learning rate	1.0	0.1	0.01	0.001
F1 score	0.5623	0.5448	0.3011	0.1

(2) n_estimator
Learning_rate = 1.0

Learning rate	100	200	500	1000
F1 score	0.5623	0.5688	0.5892	0.592