COMP 4434 Final Presenstation

JIANG Ruixiang

19079662D

Contents

Introduction

Preprocessing

Model

Result

Background

- PolyTube
- Large dataset

Project Objective

- Task 1 build a regression model
- Task 2 build a recommender system

Contents

Introduction

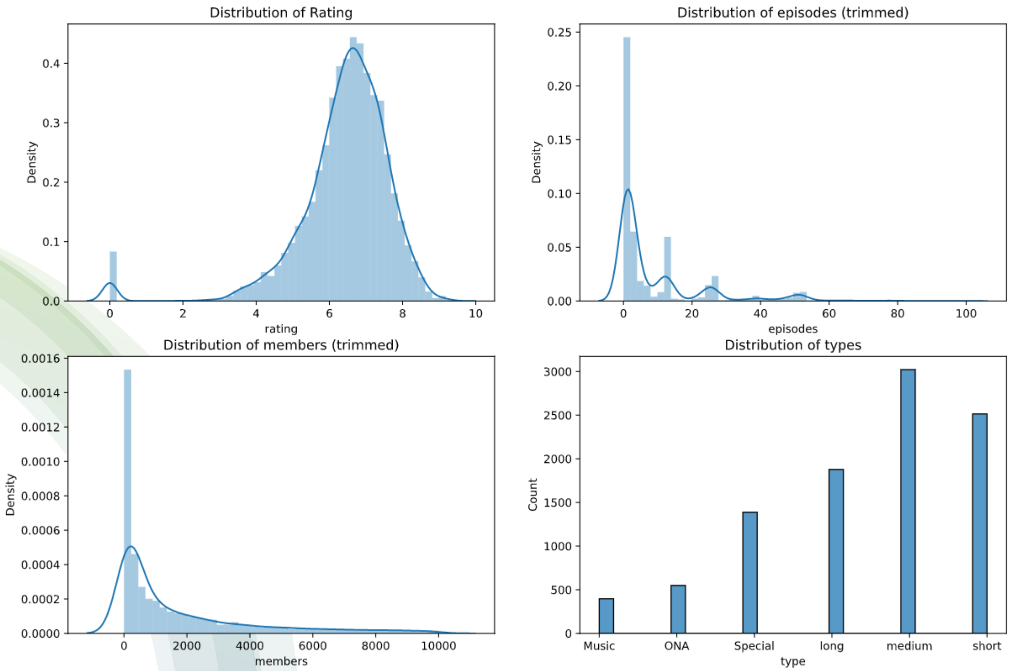
Preprocessing

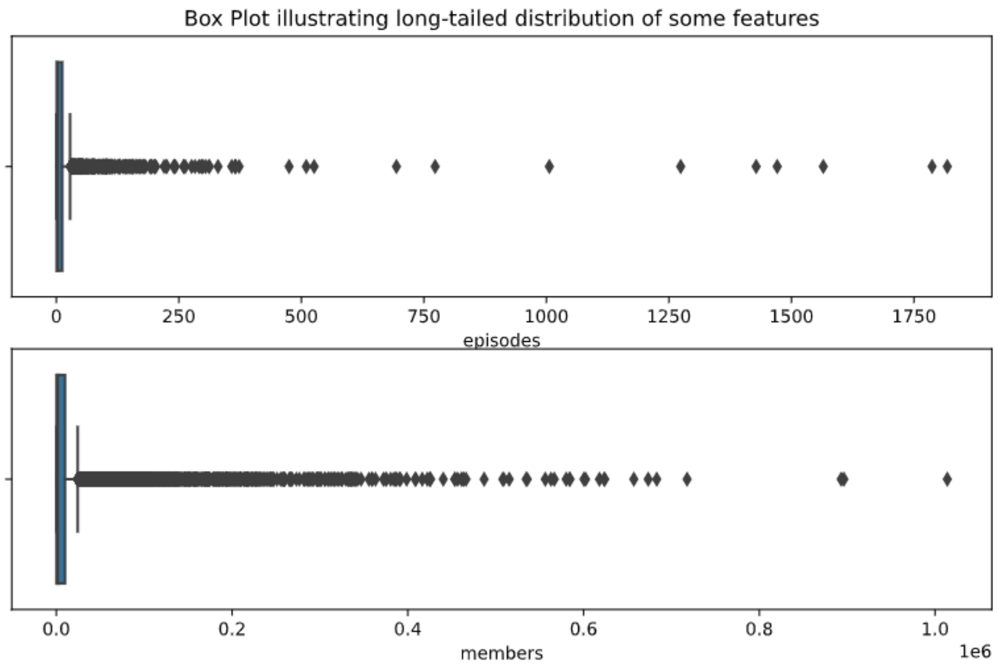
Model

Result

Exploratory data analysis

Long-tail distribution
Outliers

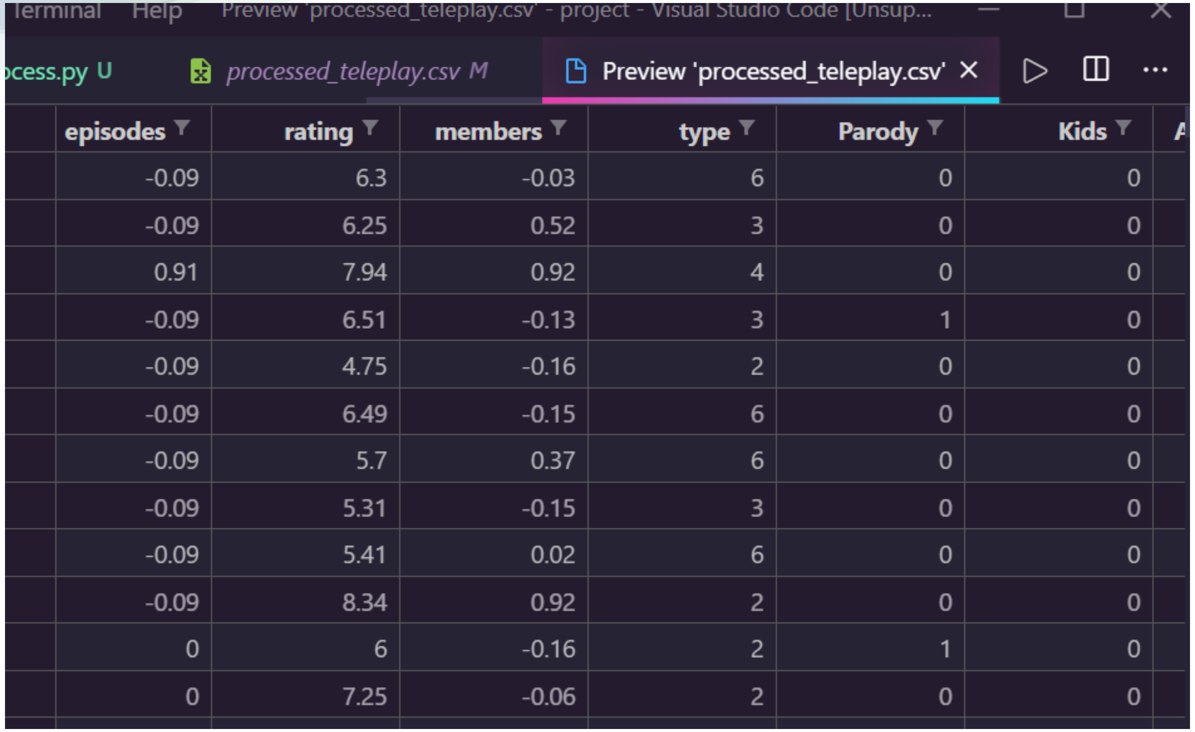




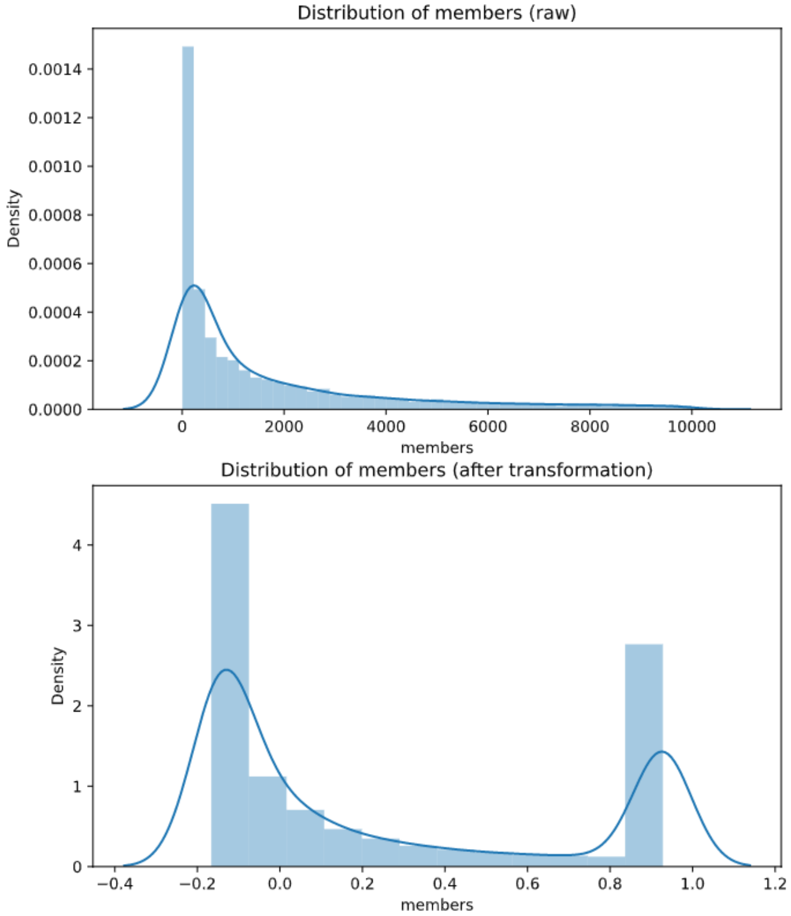
Preprocessing

- Null value imputation –mean values
- Outliers -clipping
- Feature scaling –transformation
- Feature encoding -vectorize

Preprocessing Result

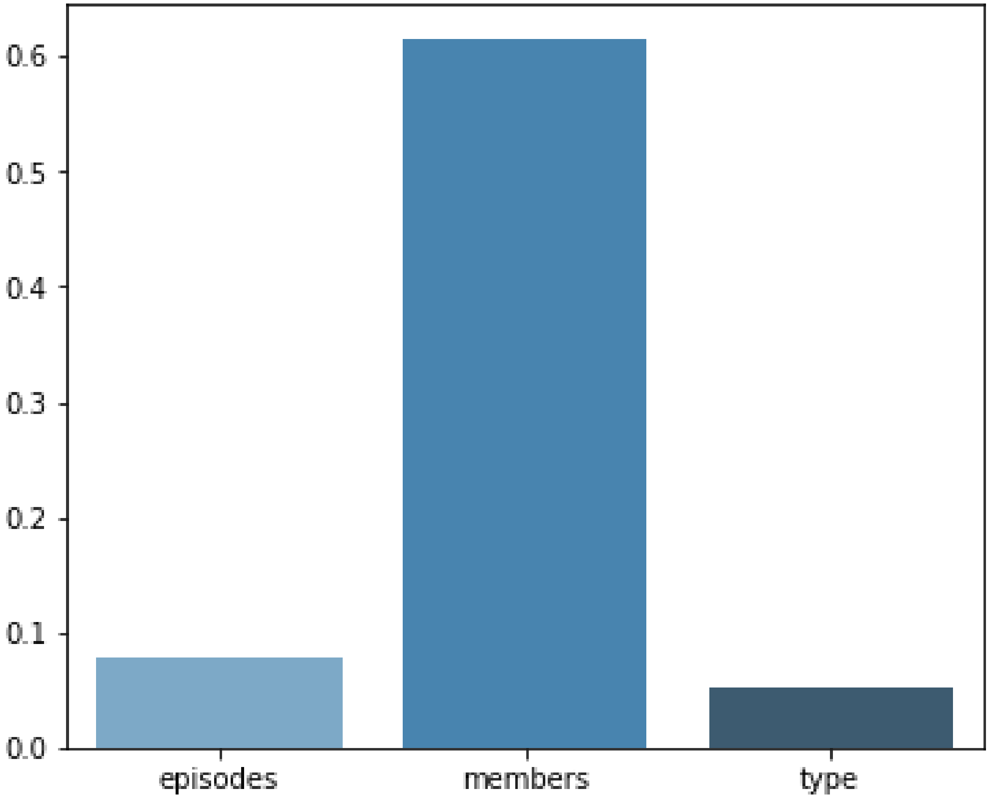

Terminal
Help
Preview 'processed_teleplay.csv' - project - Visual Studio Code [Unsup...
ocess.py U
processed_teleplay.csv M
Preview 'processed_teleplay.csv'

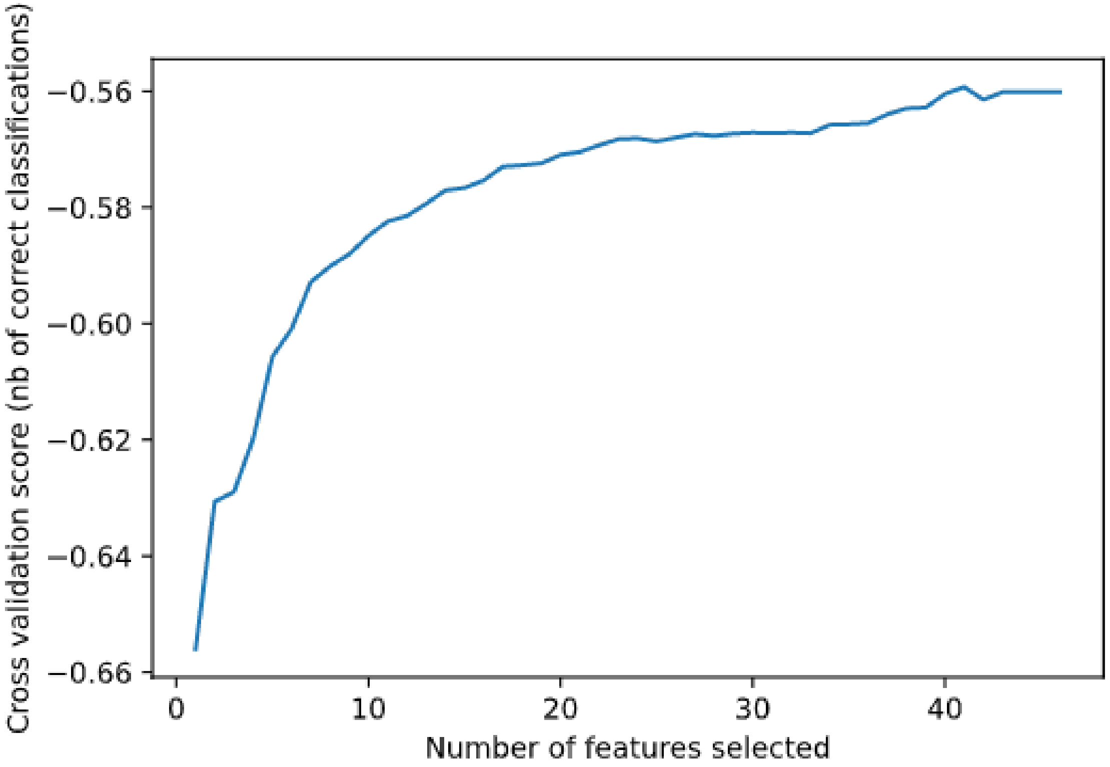
episodes	rating	members	type	Parody	Kids
-0.09	6.3	-0.03	6	0	0
-0.09	6.25	0.52	3	0	0
0.91	7.94	0.92	4	0	0
-0.09	6.51	-0.13	3	1	0
-0.09	4.75	-0.16	2	0	0
-0.09	6.49	-0.15	6	0	0
-0.09	5.7	0.37	6	0	0
-0.09	5.31	-0.15	3	0	0
-0.09	5.41	0.02	6	0	0
-0.09	8.34	0.92	2	0	0
0	6	-0.16	2	1	0
0	7.25	-0.06	2	0	0



Feature selection

- Recursive Feature Elimination
- Feature importance





Contents

Introduction

Preprocessing

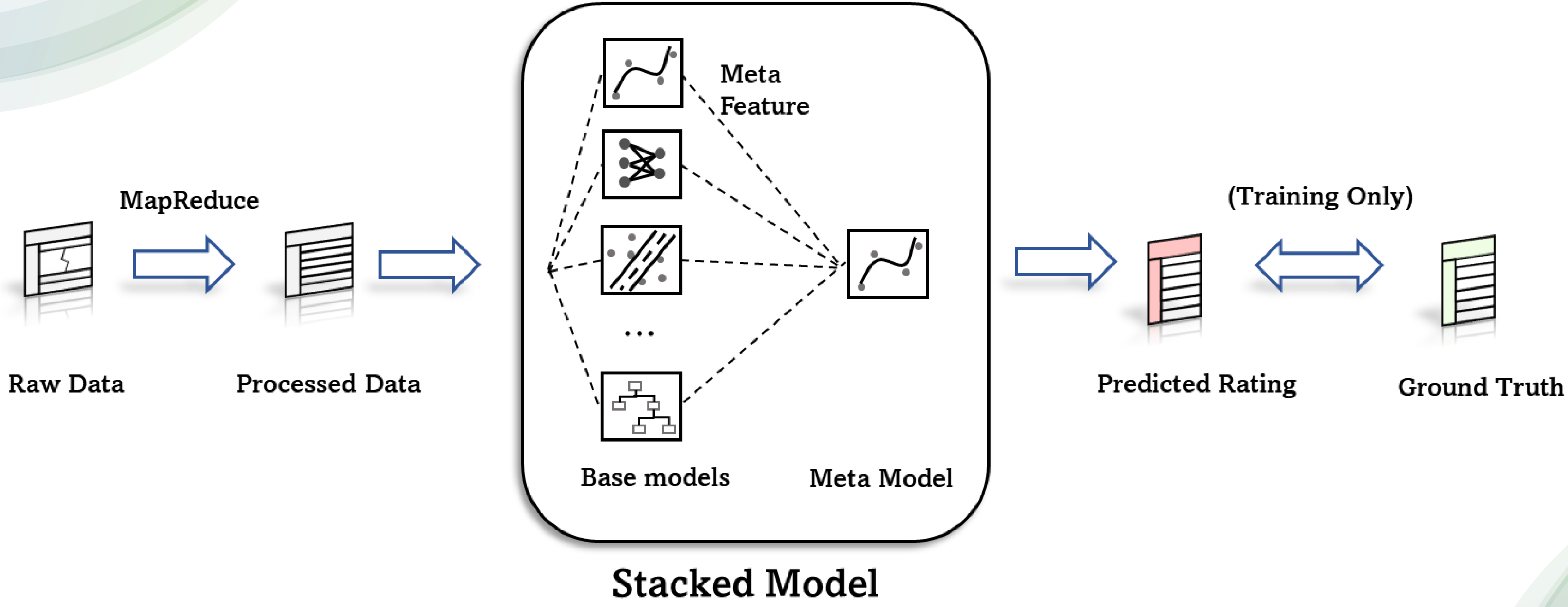
Model

Result

Model Design (Task 1)

- Baseline -linear regression, neural networks
- Ours –Stacked model (ensembling)

Model Architecture (Task 1)



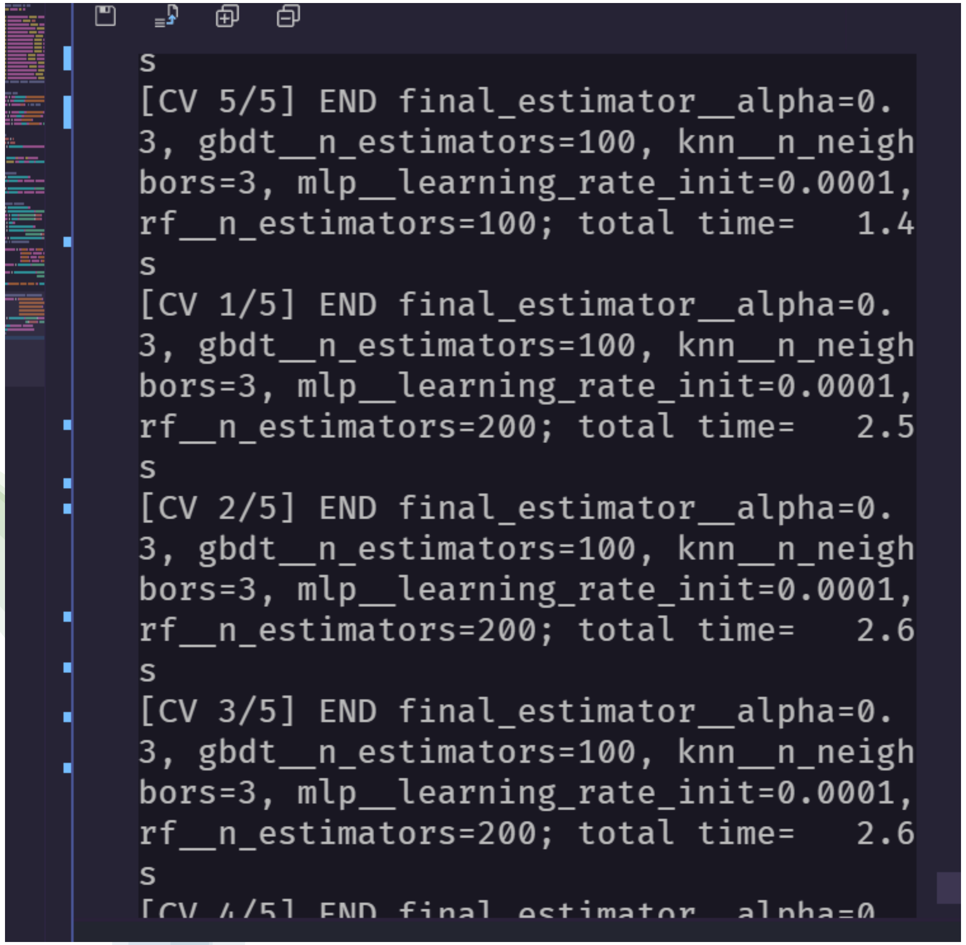
Model Design (Task 2)

- Content-based recommender system
- Many baselines
- Final solution: linear regression

Regularizations

- L1, L2 norms for LR, MLPs
- Dropout
- Batchnormalization
- Earlystopping
- reduceLRonPlatau
- Adaptive learning rate scheduler
- Gradient Clipping

Model Training



```
s
[CV 5/5] END final_estimator__alpha=0.
3, gbdt__n_estimators=100, knn__n_neigh
bors=3, mlp__learning_rate_init=0.0001,
rf__n_estimators=100; total time=   1.4
s
[CV 1/5] END final_estimator__alpha=0.
3, gbdt__n_estimators=100, knn__n_neigh
bors=3, mlp__learning_rate_init=0.0001,
rf__n_estimators=200; total time=   2.5
s
[CV 2/5] END final_estimator__alpha=0.
3, gbdt__n_estimators=100, knn__n_neigh
bors=3, mlp__learning_rate_init=0.0001,
rf__n_estimators=200; total time=   2.6
s
[CV 3/5] END final_estimator__alpha=0.
3, gbdt__n_estimators=100, knn__n_neigh
bors=3, mlp__learning_rate_init=0.0001,
rf__n_estimators=200; total time=   2.6
s
[CV 4/5] END final_estimator__alpha=0
```

- Grid Search to optimization hyper-param
- 9-12 hours on a server to optimize
- Singel training round ~10 min

Contents

Introduction

Preprocessing

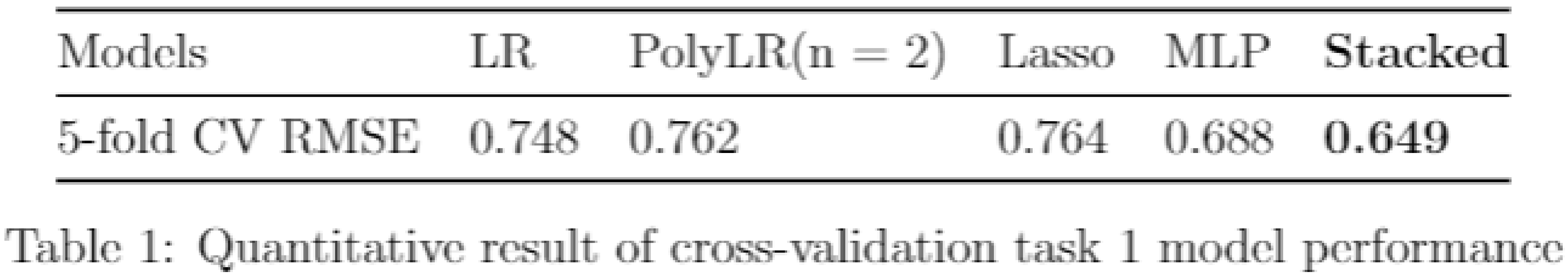
Model

Result

Model Merics

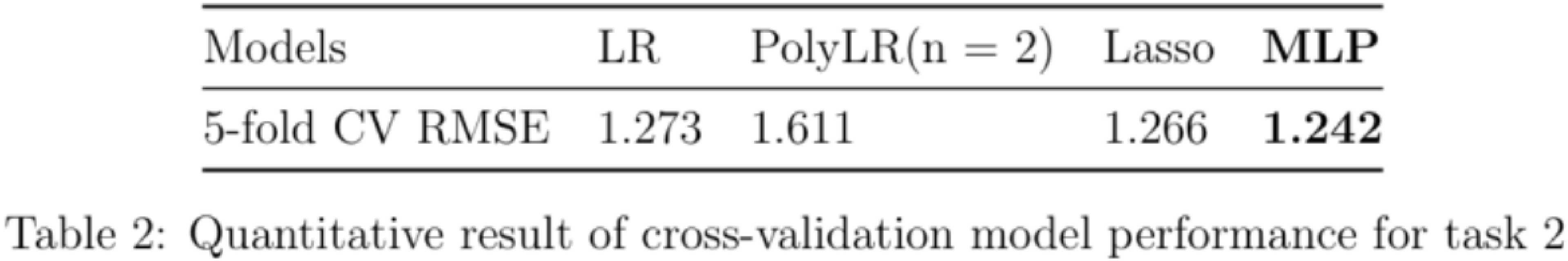
- Baselines : plain linear regression, linear regression with polynomial feature expansion, Lasso regression (λ = 0.01), multilayer perceptron
- Metrics: 5-fold RMSE

Quantitative Results



Models	LR	PolyLR(n = 2)	Lasso	MLP	**Stacked**
5-fold CV RMSE	0.748	0.762	0.764	0.688	**0.649**

Table 1: Quantitative result of cross-validation task 1 model performance



Models	LR	PolyLR(n = 2)	Lasso	**MLP**
5-fold CV RMSE	1.273	1.611	1.266	**1.242**

Table 2: Quantitative result of cross-validation model performance for task 2

Discussion on models

- Non-linear model outperform others, but training is hard
- Linear model is good for task 2
- Conclusion

Thanks