Advertisement CTR Prediction

Present Solution

DS5220 / Fall 2023 Semester

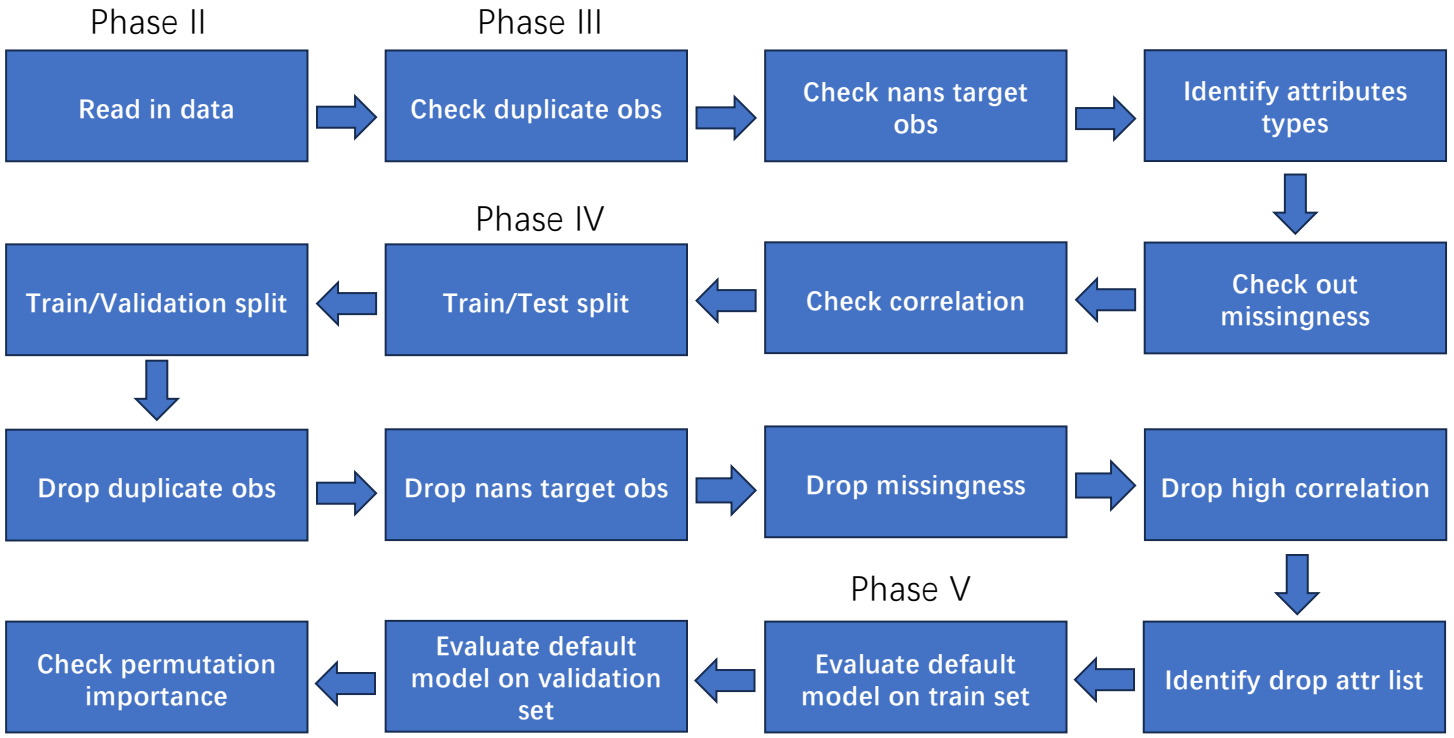
Team Members: Liyang Song, Qian Yin

Dec 10, 2023

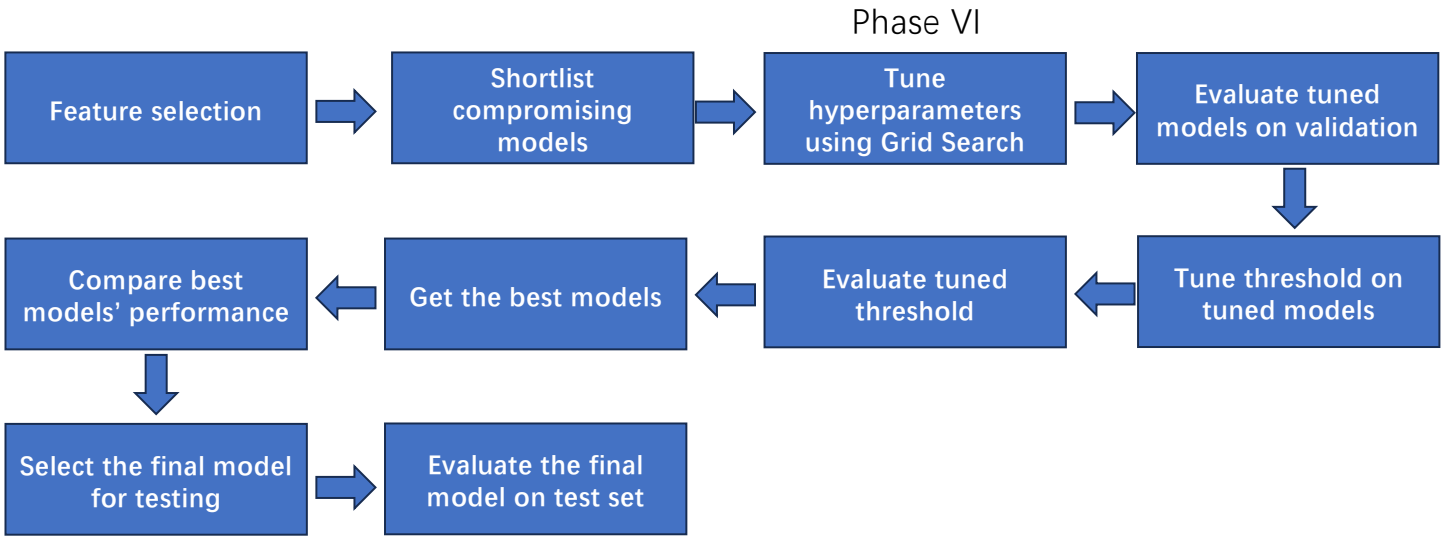
Background – big picture

- Initially, the goal is to increase the efficiency of targeted advertising by **accurately predicting the likelihood of a user clicking on an advertisement**, enabling improved return on investment (ROI) and lower cost on advertising. Therefore, we want to find the best model for the CTR (Click-Through Rate) prediction.
- There are many performance metrices to evaluate the model. In our project, we decide to focus on minimizing false positives while balancing with other metrics.
- We mainly consider **precision** or **false positive** rather than false negative because the project target is recommending advertisements to users based on the click rate prediction. As a result, higher false positive would show more advertisements that the user would be less likely to click. And this could make the user feel less engaged and tend the leave the applicaton.

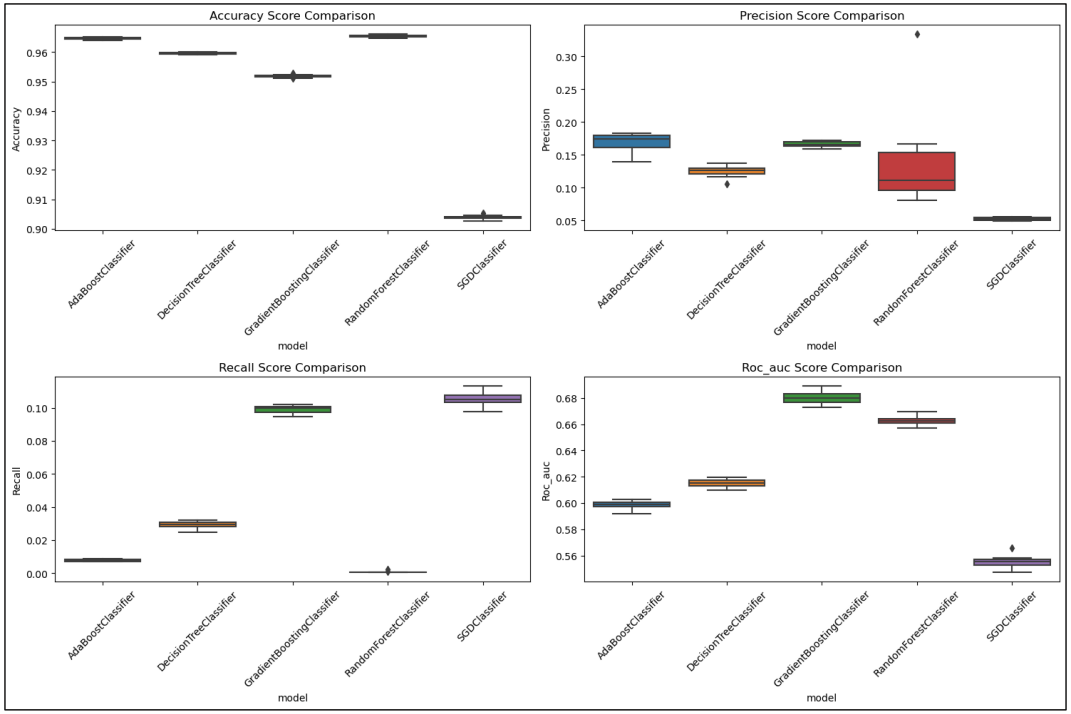
What we have done – the ML workflow



What we have done – the ML workflow (continued)



Our solution – the final best model

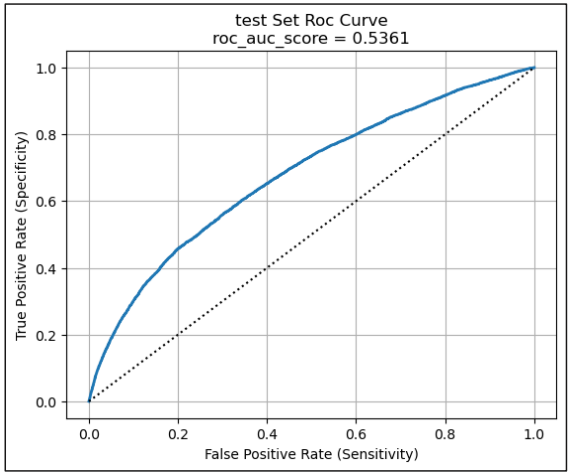


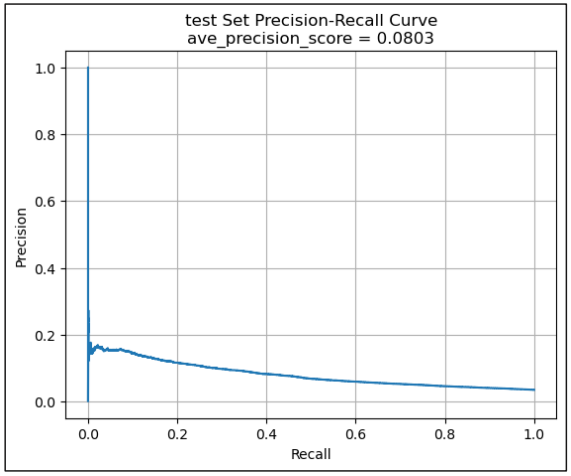
- After tuning the model hyperparameters and threshold for each best model. We evaluated them on bootstrapped validation set.
- **Gradient Boosting Classifier** has very high Precision (which is what we demands firstly) and a very high recall. It would be the ideal model for production.

Our solution – the final best model

```
Check confusion matrix
test set confusion matrix:
[[198561   3748]
 [  6571    656]]
True Positives =  198561
True Negatives =  656
False Positives(Type I error) =  3748
False Negatives(Type II error) =  6571
```

	stage	accuracy	precision	recall
0	test	0.9508	0.148955	0.090771





What worked and limitations

- Worked
  - We successfully evaluate 5 different models from various categories. They perform differently on the data set and have different error tendencies.
  - We split the raw data into train, validation, and test sets. We use the validation set to tune hyperparameters and thresholds, and use test set only at the final step to see the performance. This makes sure there is no data leakage.
  - We try to automatic almost all workflows, and most model configurations are listed at the beginning cells of notebooks. Configuring them can be very easy and straightforward.
- Limitations
  - The data set is too huge and can be hard to train and fit on our personal laptops. We have to resample a small part of the raw data set such that our work can be done.
  - The raw data is highly imbalanced, and we decided to oversample it at the final stage. Which makes the whole workflow has to be run and evaluated again. This makes our time very limited.