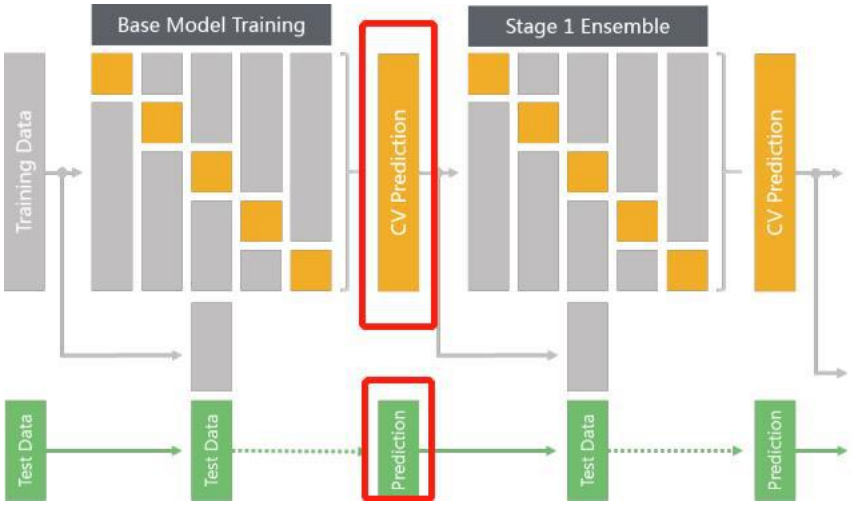
RUNNING INSTRUCTION

Simply running the **run.ipynb** can generate the submit result. It is on the google colab environment (but I suppose it can be run on other environments, too).

I use two-stage training. The other notebooks generate intermediate csv results(they are all on the kaggle platform). I save those csv results as well as the model I created. second_stage_xgboost.ipynb uses those intermediate csv results to train a xgboost model for the final prediction.

HOW I GOT THE RESULT



I use stacking for second-stage training.

Basic model training I (as shown in the left half of the first row in Figure).
According to the cross validation method, the model is trained on the training fold (as shown in the gray part of the figure), and the prediction is made on the validation fold to get the prediction result (as shown in the yellow part of the figure). Finally, the prediction results of the whole training set are merged (as shown in the CV prediction of the first yellow part of the figure).

Basic model training II (as shown in the left half of the second and third rows in Figure).
Train the model on the full training set (as shown in the gray part of the second line of the figure), and make predictions on the test set to get the prediction results (as shown in the green part after the dotted line in the third line).

I trained those 4 base models independently using the categorical /image/text data:
1. categorical data: **xgboost**

2. image data: **resnet18**
   For the scheduler, I choose Cosine Annealing with Restart
3. text data: **fast-text**
4. text data: **lstm**

I got one predicted training set and one predicted test set for intermediate results (see the red rectangle in Figure). I merge those results to one data frame. Since I trained four baseline model, it has 4*27 columns.
For the second stage training, I use xgboost to get the final prediction.
The final prediction accuracy is around 97.7%.