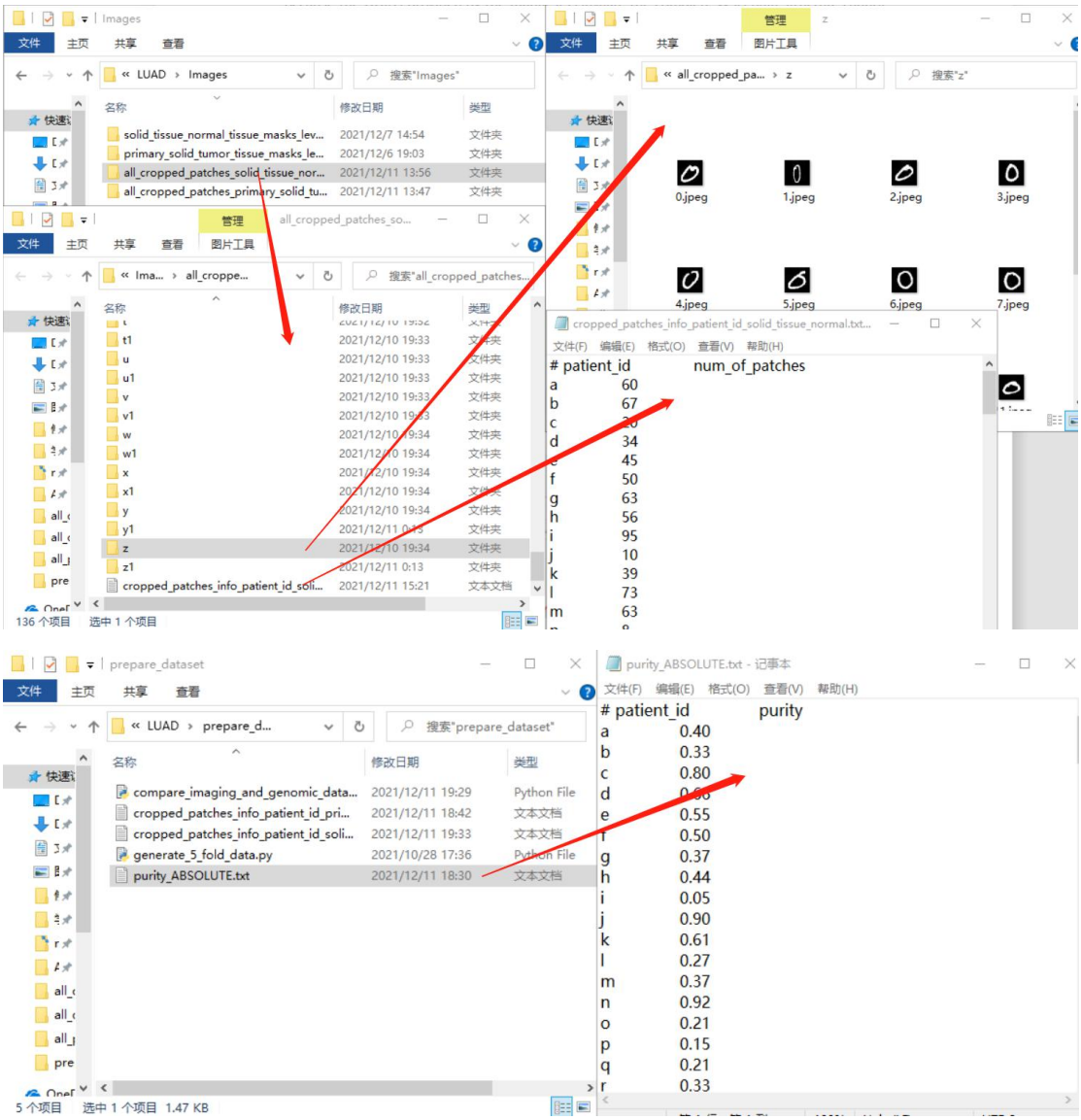
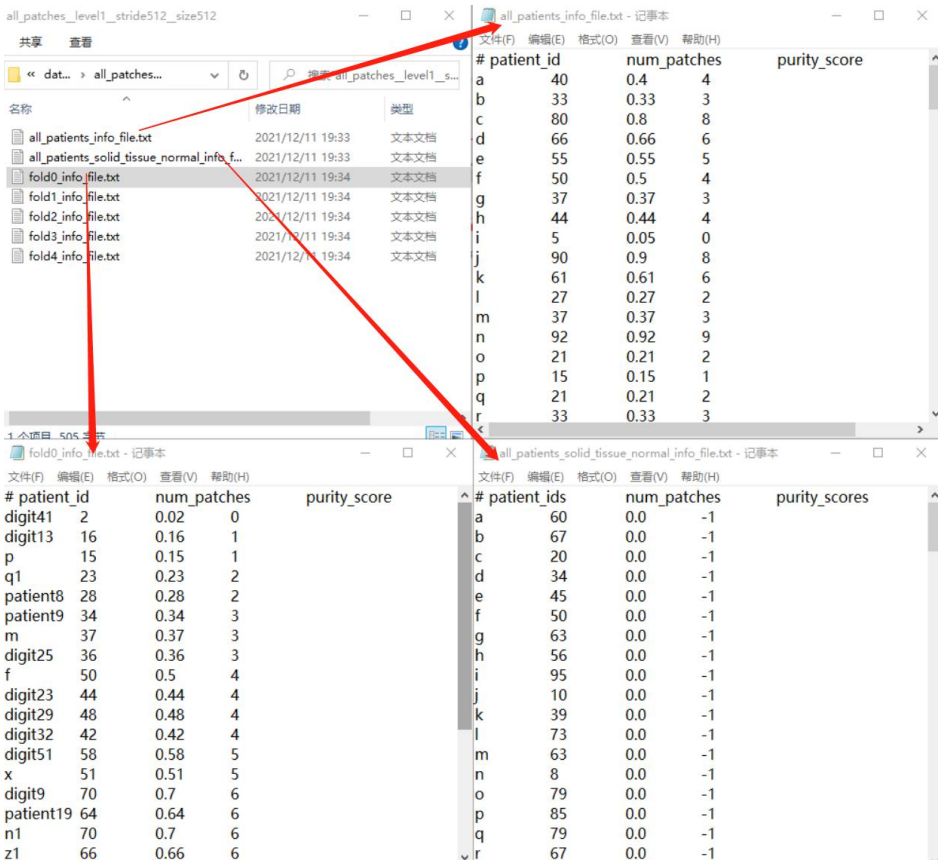
Q1

As mentioned in my previous submission, I encountered a problem with the program inputting pictures. After careful investigation, I found the simple but fatal error: the program that automatically modifies the name of the picture added a blank space before the order number, so that the picture could not be input. This error wasted a lot of my time. After I corrected it, the next work went well, and I will introduce it again.

Because the UUIDs provided by the manifest is invalid, the complete WSIs data structure cannot be obtained. Therefore, I construct my own MNIST dataset according to the *Tissue Masks and Patch Cropping*.I set 135 bags. The number of 1 and 7 in each bag adds up to 100, and two folders are set for two numbers respectively, and the corresponding TXT file has been written:



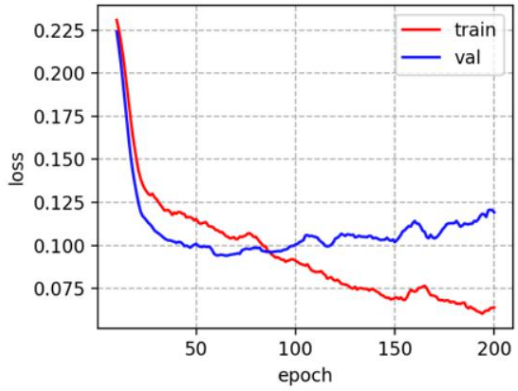
Then I run *compare_imaging_and_genomic_data.py* and *generate_5_fold_data.py*, *purity_score* was successfully obtained and the data were divided into 5 groups with 27 ids in each group:

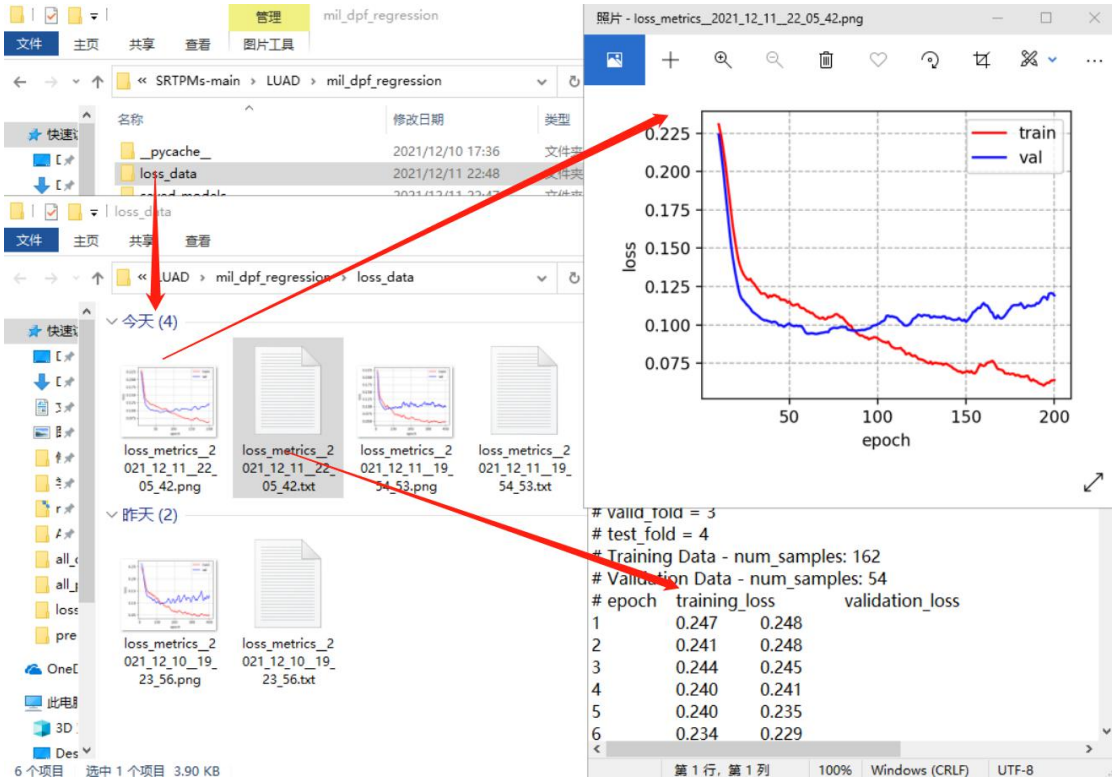


Next I run *train.py*,I change *--patch_size* to 28, *--num_instances* to 100, *--batch_size* to 64, *--num_epochs* to 200, *--save_interval* to 2 and *num_workers* to 0:

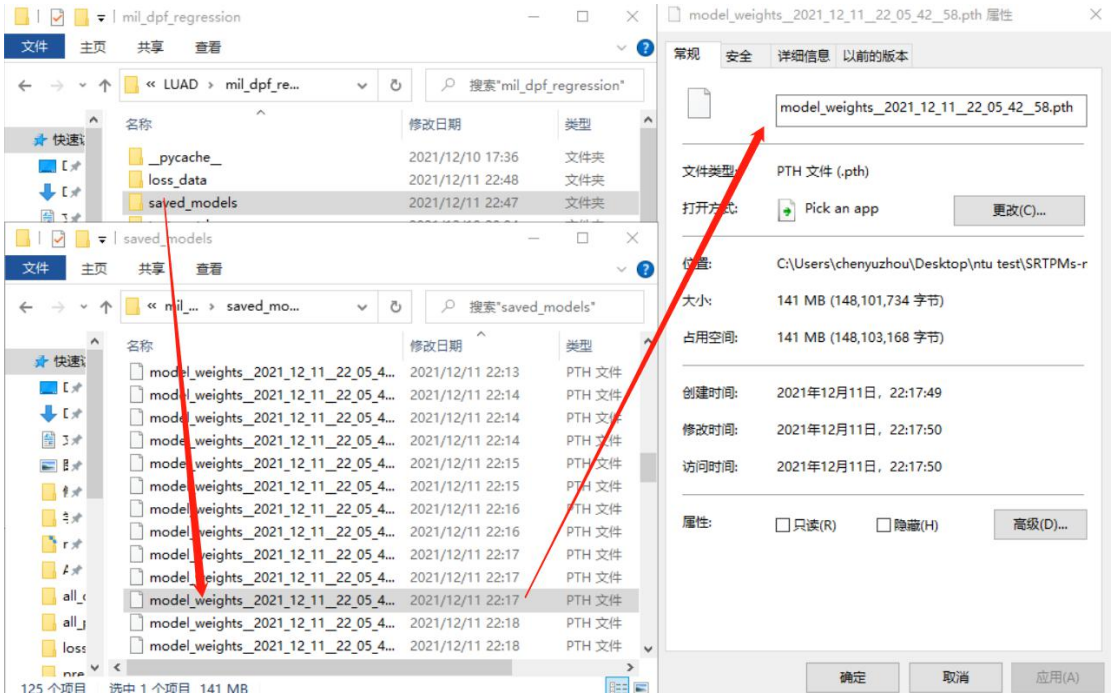
```
parser.add_argument('--init_model_file', default='', help='the path of initial model file', dest='init_model_file')
parser.add_argument('--image_dir', default='../Images/all_cropped_patches_primary_solid_tumor__level1__stride512__size512', help='Image directory for tumor patches', d
parser.add_argument('--normal_image_dir', default='../Images/all_cropped_patches_solid_tissue_normal__level1__stride512__size512', help='Image directory for normal pat
parser.add_argument('--dataset_dir', default='../dataset/all_patches__level1__stride512__size512', help='dataset info folder', dest='dataset_dir')
parser.add_argument('--patch_size', default='28', type=int, help='patch size', dest='patch_size')
parser.add_argument('--num_instances', default='100', type=int, help='number of instances (patches) in a bag', dest='num_instances')
parser.add_argument('--num_features', default='128', type=int, help='number of features', dest='num_features')
parser.add_argument('--num_bins', default='21', type=int, help='number of bins in distribution pooling filter', dest='num_bins')
parser.add_argument('--sigma', default='0.05', type=float, help='sigma in distribution pooling filter', dest='sigma')
parser.add_argument('--num_classes', default='1', type=int, help='number of classes', dest='num_classes')
parser.add_argument('--batch_size', default='64', type=int, help='batch size', dest='batch_size')
parser.add_argument('--learning_rate', default='1e-4', type=float, help='number of patches each patient has', dest='learning_rate')
parser.add_argument('--num_epochs', default=400, type=int, help='number of steps of execution (default: 1000000)', dest='num_epochs')
parser.add_argument('--save_interval', default=16, type=int, help='model save interval (default: 1000)', dest='save_interval')
parser.add_argument('--metrics_dir', default='loss_data', help='file to log training metrics (e.g. loss)', dest='metrics_dir')
parser.add_argument('--models_dir', default='saved_models', help='directory to save models', dest='models_dir')
parser.add_argument('--valid_fold', default=3, type=int, help='id of fold to be used as validation set', dest='valid_fold')
parser.add_argument('--test_fold', default=4, type=int, help='id of fold to be used as test set', dest='test_fold')
```

Result:

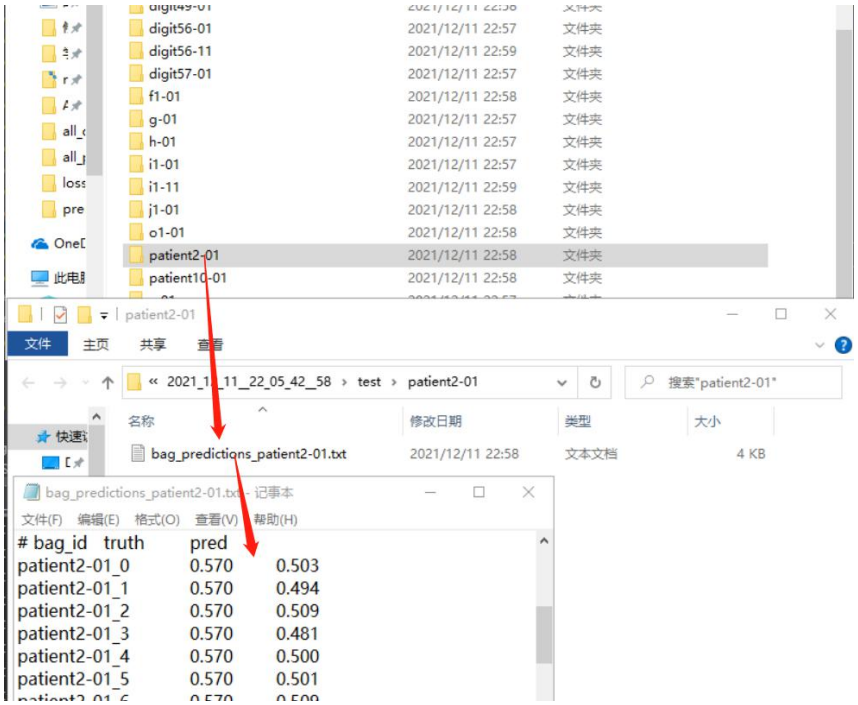




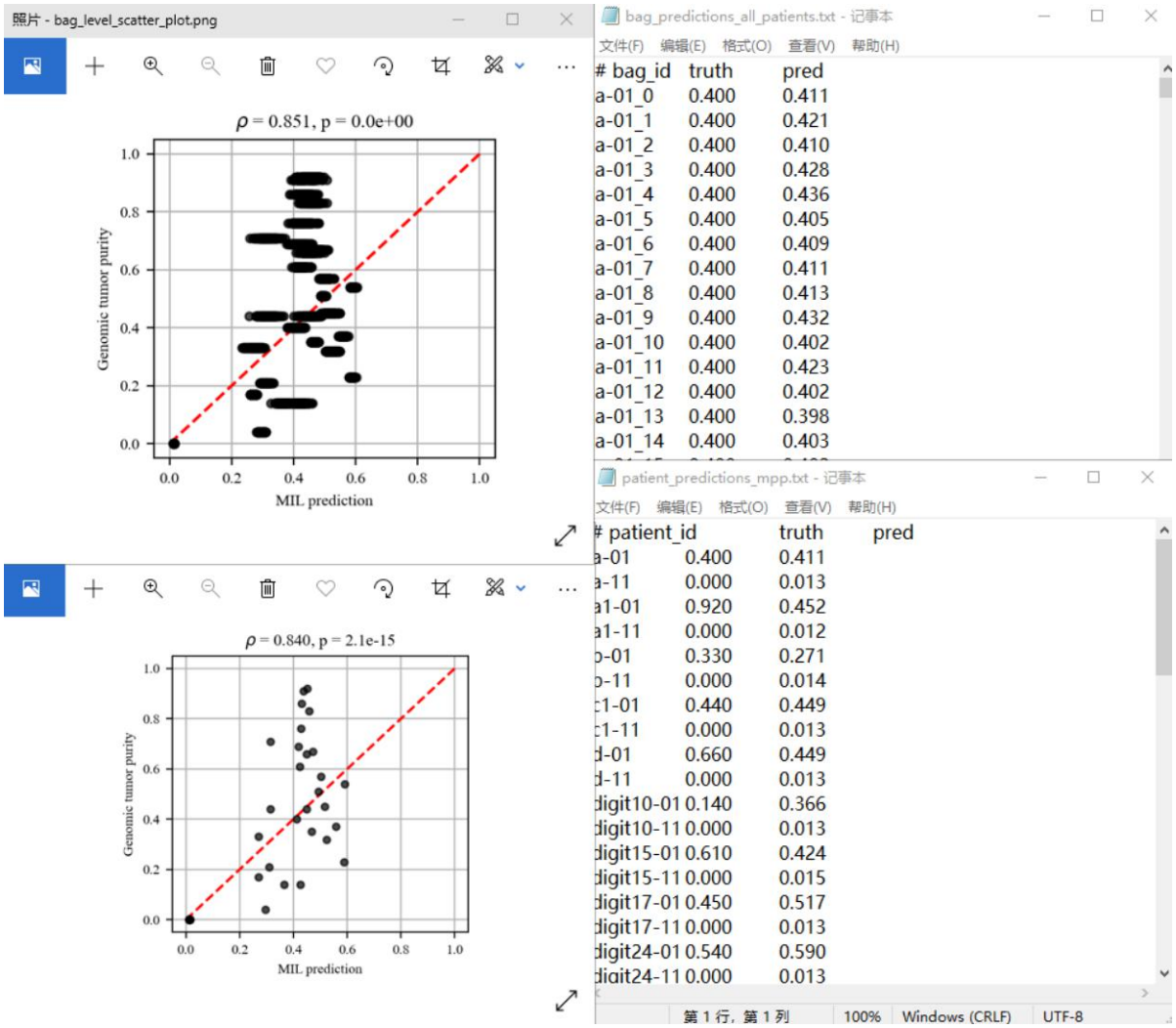
It can be seen that the model has the best generalization when *epoch* is around 58, so I chose *model_weights__2021_12_11__22_05_42__58.pth* for the follow-up test.



To obtain sample-level tumor purity predictions from a trained model, run *test_patient.py*:



To obtain sample-level predictions,run *collect_statistics_over_bag_predictions__sample_level.py*:

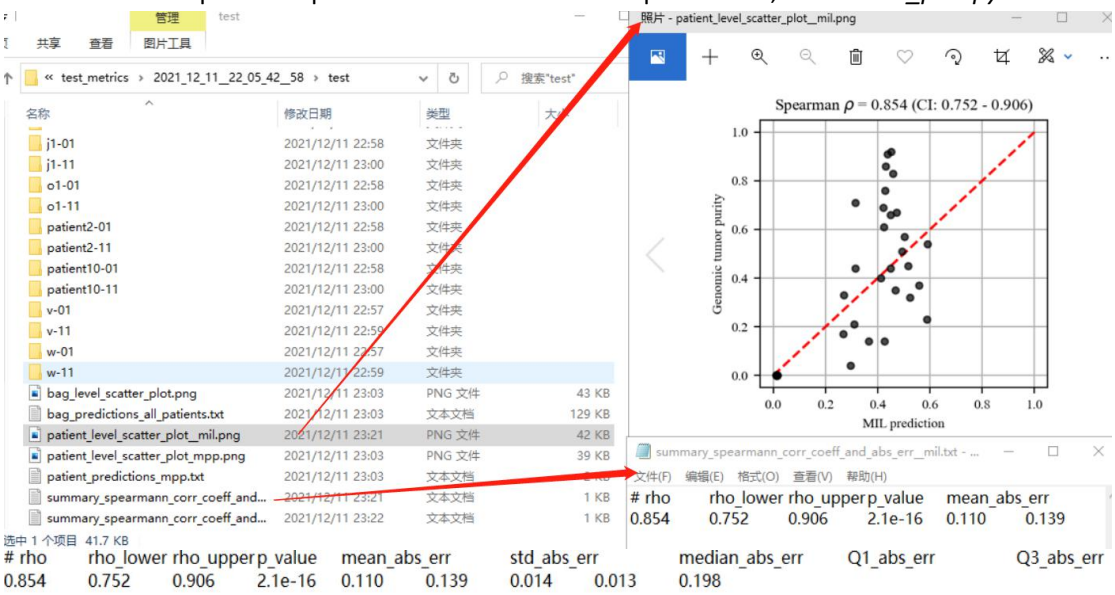


We can see that bag_level *genomic tumor purity* is evenly distributed, while *MIL prediction* is concentrated within 0.3 and 0.6, indicating that the prediction ability of the model is not ideal.
On the other hand, the Spearman's coefficient is about 0.85, which means the two have strong monotonicity.The P value is less than 0.05, which is statistically significant.

Because of the lack of *sample_id__analyte_portion_id__percent_tumor_nuclei.txt*, we can't collate *pathologists' estimates and the MIL model's predictions*:

```
    raise IOError("%s not found." % path)
OSError: ../tcga_data/sample_id__analyte_portion_id__percent_tumor_nuclei.txt not found.
```

To obtain scatter plots and performance metrics for MIL predictions, run *scatter_plot.py*:



# rho	rho_lower	rho_upper	p_value	mean_abs_err		std_abs_err		median_abs_err	Q1_abs_err	Q3_abs_err
0.854	0.752	0.906	2.1e-16	0.110	0.139	0.014	0.013	0.198		

Because of the limitation of the conditions, I only constructed the sample-level data, so the results of slide-level cannot be obtained.

Judging from the current results, the model is more inclined to give a moderate purity prediction. However, there is a strong monotonicity between the model's prediction result and the real value.