4. Handle Ordinal Data

(Before analysis)Decisions I made:

1). I make all the instances as both training set and test set

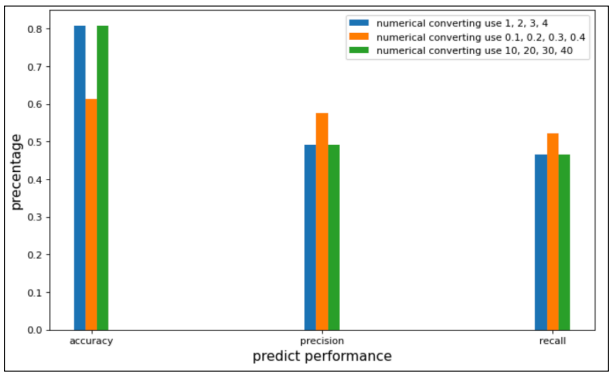
I tried both using no test data(use all data for both training and evaluating) and splitting data in to 80% training and 20% testing. The results are quit similar, and using no test data can see the performance of the model more clearly.

```
test for Nursery Dataset (no test data)

accuracy is: 0.903009
error rate is: 0.096991

----------------the macro averaging way------------------
Marco averaging precision is:  0.7248691953671452
Marco averaging recall is: 0.566428373344821
----------------the micro averaging way------------------
Mirco averaging precision is: 0.9030092592592592
Mirco averaging recall is: 0.9030092592592592
----------------the weight averaging way------------------
weight averaging precision is: 0.9056265890564255
weight averaging recall is: 0.9030092592592593
```

```
test for Nursery Dataset (split data into 80% for training and 20% for test)

accuracy is: 0.896605
error rate is: 0.103395

----------------the macro averaging way------------------
Marco averaging precision is:  0.740087597409457
Marco averaging recall is: 0.5652675395545248
----------------the micro averaging way------------------
Mirco averaging precision is: 0.8966049382716049
Mirco averaging recall is: 0.8966049382716049
----------------the weight averaging way------------------
weight averaging precision is: 0.9029365999980138
weight averaging recall is: 0.896604938271605
```

(Evaluation result of nursery.data)

2). I map the original data to integer 1,2,3,4 according to its order



Result of different converting numeric value of car.data

Cause all the ordinal data has its own order information, the value it mapping should be according to its order.

At the same time, in the ordinal datasets in our assignment. The value that each attribute has are few. So using integer is good.

Also I compare the result to map value to float number, single digit integer, Two-digit integer.The result show the integer gives the highest accuracy. So using integer with gap 1 is the best choice

For example, in car.data, attribute buying, I mapped “v_high”->3, “high"->2, “med"->1, “low”->0

Analysis:

First Half Question (which approach has higher classification accuracy)

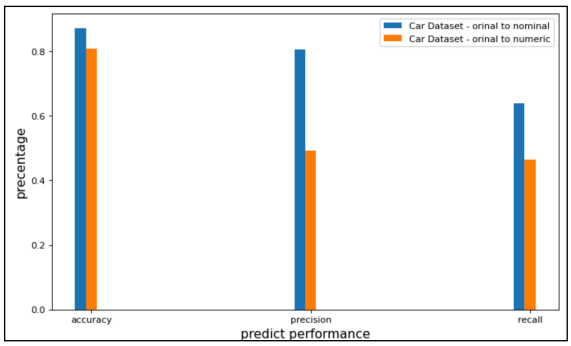
For all the datasets that this assignment provide, there are three ordinal attributes only datasets, which is car.data, nursery.data, somerville.data. (There also some datasets with mix type attribute with ordinal data inside,In order to make the effect of ordinal most obvious, I didn’t use them).

I convert these three datasets values into both nominal variables and numeric variables, and run the Navie Bayes classification on each of them.

By comparing the result, we can say that converting ordinal variables into the nominal variables is the better choice.

The result of car.data

```
################ Car Dataset - orinal to nominal result ################
accuracy is: 0.871528
error rate is: 0.128472

----------------the macro averaging way------------------
Marco averaging precision is:  0.8040352387146505
Marco averaging recall is: 0.6378405556688042
----------------the micro averaging way------------------
Mirco averaging precision is: 0.8715277777777778
Mirco averaging recall is: 0.8715277777777778
----------------the weight averaging way------------------
weight averaging precision is: 0.8688062641518524
weight averaging recall is: 0.8715277777777779
```

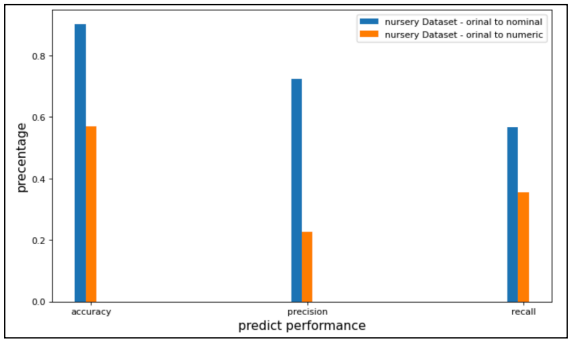



```
################ Car Dataset - orinal to numeric result ################
accuracy is: 0.808449
error rate is: 0.191551

----------------the macro averaging way------------------
Marco averaging precision is:  0.4911436895877529
Marco averaging recall is: 0.46454782196969696
----------------the micro averaging way------------------
Mirco averaging precision is: 0.8084490740740741
Mirco averaging recall is: 0.8084490740740741
----------------the weight averaging way------------------
weight averaging precision is: 0.7668494144356257
weight averaging recall is: 0.8084490740740741
```

The accuracy of nominal one is higher than the numerical one

The result of nursery.data

```
################ nursery Dataset - orinal to nominal result ################
accuracy is: 0.903009
error rate is: 0.096991

----------------the macro averaging way------------------
Marco averaging precision is:  0.7248820267369591
Marco averaging recall is: 0.5664257996860103
----------------the micro averaging way------------------
Mirco averaging precision is: 0.9030092592592592
Mirco averaging recall is: 0.9030092592592592
----------------the weight averaging way------------------
weight averaging precision is: 0.905644265528916
weight averaging recall is: 0.9030092592592592
```

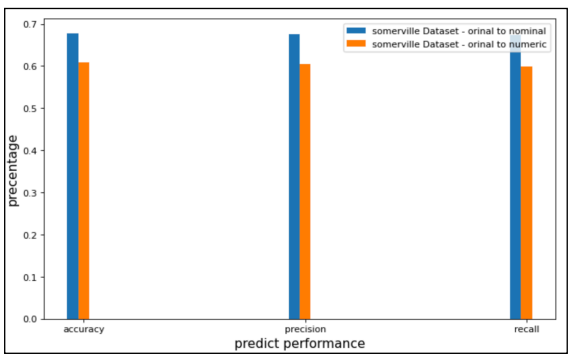



```
################ somerville Dataset - orinal to numeric result ################
accuracy is: 0.608392
error rate is: 0.391608

----------------the macro averaging way------------------
Marco averaging precision is:  0.6052850603412401
Marco averaging recall is: 0.5995670995670996
----------------the micro averaging way------------------
Mirco averaging precision is: 0.6083916083916084
Mirco averaging recall is: 0.6083916083916084
----------------the weight averaging way------------------
weight averaging precision is: 0.6062614040142129
weight averaging recall is: 0.6083916083916083
```

The accuracy of nominal one is higher than the numerical one. And It is much higher, nearly 30%

The result of somerville.data

```
################ somerville Dataset - orinal to nominal result ################
accuracy is: 0.678322
error rate is: 0.321678

----------------the macro averaging way------------------
Marco averaging precision is:  0.6763241736360015
Marco averaging recall is: 0.6742424242424243
----------------the micro averaging way------------------
Mirco averaging precision is: 0.6783216783216783
Mirco averaging recall is: 0.6783216783216783
----------------the weight averaging way------------------
weight averaging precision is: 0.6774806237171829
weight averaging recall is: 0.6783216783216783
```



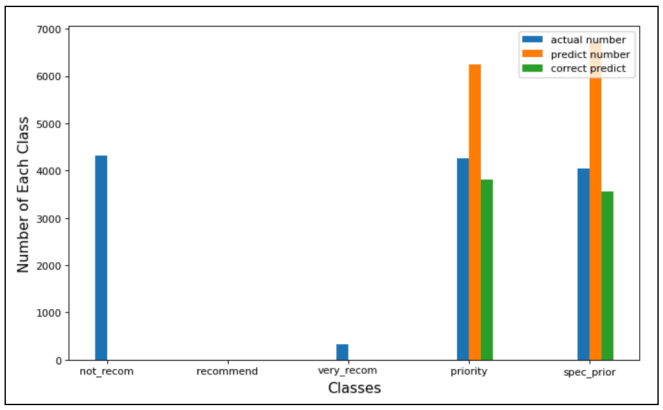

```
################ somerville Dataset - orinal to numeric result ################
accuracy is: 0.608392
error rate is: 0.391608

----------------the macro averaging way------------------
Marco averaging precision is:  0.6052850603412401
Marco averaging recall is: 0.5995670995670996
----------------the micro averaging way------------------
Mirco averaging precision is: 0.6083916083916084
Mirco averaging recall is: 0.6083916083916084
----------------the weight averaging way------------------
weight averaging precision is: 0.6062614040142129
weight averaging recall is: 0.6083916083916083
```

The accuracy of nominal one is still higher than the numerical one.

According to these three comparison pair, We can see that the accuracy of converting to nominal is always higher than converting to numeric. And it is higher about 7% to 30%. So treating the ordinal data as the nominal data is the approach has the higher classification accuracy.

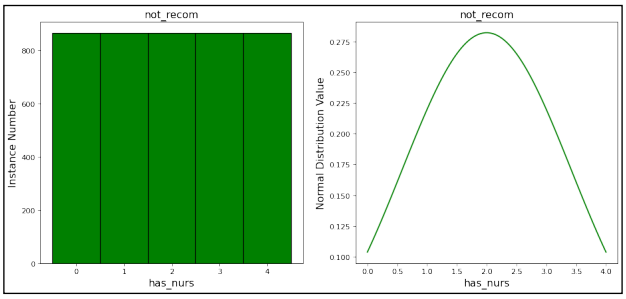
Second Half Question (why)

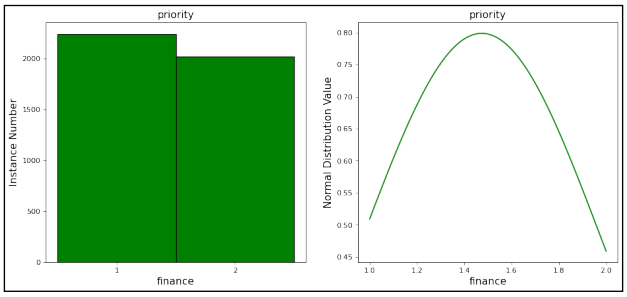
Because, in the nursery.data's, the difference between the two way is the biggest. So I read into the training and testing process.

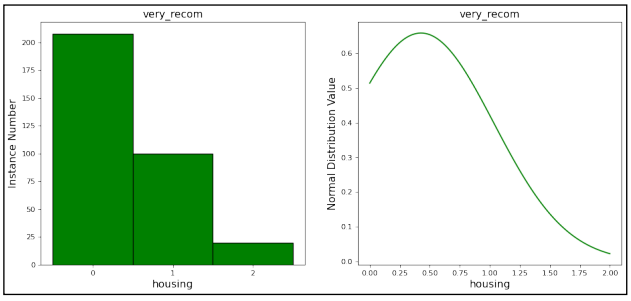


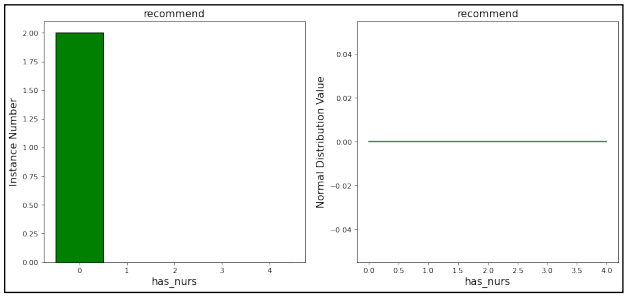
When we print out the number of different class(the actual number, predict number, and the correctly predict number). We can see that for class not_recom, recommend and very_recom, our model predicts none of them.

And when we print out the distribution of the attribute, we can find out the reason









As we can see the the distribution of the attributes are not normal distributed.
Like the has_nurs attribute under not_recom class, and finance attribute under the priority class are the uniform distribution.
Even the housing attribute under very_recom class, it has kind of normal distribution shape, but there are not enough type of value.
And the has_nurs under recommend class, only have only value.

So when we use the normal distribution method to calculate its probability, the result will be not accurate.

Because the ordinal attribute has few values, when it converts to numerical value, it will not match the normal distribution very good. To that extend, The accuracy will be lower than converting to the nominal values.