In homework 2, I implement a decision tree and Naïve Bayes classifier for classification. The decision tree models are built using two criteria: 'Gini' and 'entropy'. Different types of Naïve Bayes models, such as Bernoulli, Gaussian, and others are built concerning specific types of datasets. The data processing procedures are specified as follows.

First, the values of the house-votes dataset are turned into numeric data types, with "republican", "democrat", "y", and "n" replaced by 1, 0, 1, 0 respectively.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
0	1	0.0	1.0	0.0	1.0	1.0	1.0	0.0	0.0	0.0	1.0	NaN	1.0	1.0	1.0	0.0	1.0
1	1	0.0	1.0	0.0	1.0	1.0	1.0	0.0	0.0	0.0	0.0	0.0	1.0	1.0	1.0	0.0	NaN
2	0	NaN	1.0	1.0	NaN	1.0	1.0	0.0	0.0	0.0	0.0	1.0	0.0	1.0	1.0	0.0	0.0
3	0	0.0	1.0	1.0	0.0	NaN	1.0	0.0	0.0	0.0	0.0	1.0	0.0	1.0	0.0	0.0	1.0
4	0	1.0	1.0	1.0	0.0	1.0	1.0	0.0	0.0	0.0	0.0	1.0	NaN	1.0	1.0	1.0	1.0

Then, generate three datasets that handle missing values in three different ways. The first one is to "discard instances with missing values". After dropping all the instances with missing values. There are no NaN values left in the dataset. See the processed dataset below.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
5	0	0.0	1.0	1.0	0.0	1.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	1.0	1.0	1.0
8	1	0.0	1.0	0.0	1.0	1.0	1.0	0.0	0.0	0.0	0.0	0.0	1.0	1.0	1.0	0.0	1.0
19	0	1.0	1.0	1.0	0.0	0.0	0.0	1.0	1.0	1.0	0.0	1.0	0.0	0.0	0.0	1.0	1.0
23	0	1.0	1.0	1.0	0.0	0.0	0.0	1.0	1.0	1.0	0.0	0.0	0.0	0.0	0.0	1.0	1.0
25	0	1.0	0.0	1.0	0.0	0.0	0.0	1.0	1.0	1.0	1.0	0.0	0.0	0.0	0.0	1.0	1.0

The second way is to treat missing values as new values by replacing NaN with 0.5, which is the mean value of 0 and 1. The processed dataset is shown below.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
0	1	0.0	1.0	0.0	1.0	1.0	1.0	0.0	0.0	0.0	1.0	0.5	1.0	1.0	1.0	0.0	1.0
1	1	0.0	1.0	0.0	1.0	1.0	1.0	0.0	0.0	0.0	0.0	0.0	1.0	1.0	1.0	0.0	0.5
2	0	0.5	1.0	1.0	0.5	1.0	1.0	0.0	0.0	0.0	0.0	1.0	0.0	1.0	1.0	0.0	0.0
3	0	0.0	1.0	1.0	0.0	0.5	1.0	0.0	0.0	0.0	0.0	1.0	0.0	1.0	0.0	0.0	1.0
4	0	1.0	1.0	1.0	0.0	1.0	1.0	0.0	0.0	0.0	0.0	1.0	0.5	1.0	1.0	1.0	1.0

The third way is to impute missing data by replacing missing values with the most common value for that feature. The processed dataset is shown below.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
0	1	0.0	1.0	0.0	1.0	1.0	1.0	0.0	0.0	0.0	1.0	0.0	1.0	1.0	1.0	0.0	1.0
1	1	0.0	1.0	0.0	1.0	1.0	1.0	0.0	0.0	0.0	0.0	0.0	1.0	1.0	1.0	0.0	1.0
2	0	0.0	1.0	1.0	0.0	1.0	1.0	0.0	0.0	0.0	0.0	1.0	0.0	1.0	1.0	0.0	0.0
3	0	0.0	1.0	1.0	0.0	1.0	1.0	0.0	0.0	0.0	0.0	1.0	0.0	1.0	0.0	0.0	1.0
4	0	1.0	1.0	1.0	0.0	1.0	1.0	0.0	0.0	0.0	0.0	1.0	0.0	1.0	1.0	1.0	1.0

Next, let's first train the decision tree classification model. The precision, recall, and f1 scores are calculated and the performance can be evaluated. The following shows all the training results with two different criteria applied.

```
Decision Tree (gini) (dicard missing values):
    Avarage precision score: 0.942462962962963
    Avarage recall score: 0.9480185185185185
    Avarage f1 score: 0.9442248271289191

Decision Tree (entropy) (dicard missing values):
    Avarage precision score: 0.9327407407407406
    Avarage recall score: 0.9688518518518519
    Avarage f1 score: 0.9494004542715431

Decision Tree (gini) (ternary feature):
    Avarage precision score: 0.9126500759740306
    Avarage recall score: 0.9108165809568334
    Avarage f1 score: 0.9102520763362753

Decision Tree (entropy) (ternary feature):
    Avarage precision score: 0.9053320683111954
    Avarage recall score: 0.9233165809568333
    Avarage f1 score: 0.9125545362457377

Decision Tree (gini) (impute missing values):
    Avarage precision score: 0.9139661699050492
    Avarage recall score: 0.9072054698457223
    Avarage f1 score: 0.9094621361599291

Decision Tree (entropy) (impute missing values):
    Avarage precision score: 0.928781526223128
    Avarage recall score: 0.9233165809568333
    Avarage f1 score: 0.9249781762986637
```

From the results, we see that the models using entropy as the criterion have higher f1 scores than Gini on all types of datasets we feed in. Higher f1 scores indicate better performance since it combines the precision and recall of a classifier into a single metric by taking their harmonic mean. So, in these cases, "Entropy" is a better criterion option than "Gini".

The second classifier implemented is Naïve Bayes. Among all the data that have been processed, the one with ternary features is not in binary form, thus four types of Naïve Bayes models, including Gaussian, Categorical, and Multinomial are applied. Other than that, Bernoulli Naïve Bayes is used for the rest of the data that are in binary form. The result is shown below.

```
Bernoulli Naïve Bayes (dicard missing values):
    Avarage precision score: 0.8753767560664111
    Avarage recall score: 0.9310185185185185
    Avarage f1 score: 0.8979996038819568

Multinomial Naïve Bayes (ternary feature):
    Avarage precision score: 0.8282203907203908
    Avarage recall score: 0.9111352657004831
    Avarage f1 score: 0.8656364266136564

Categorical Naïve Bayes (ternary feature):
    Avarage precision score: 0.8199962949962952
    Avarage recall score: 0.896203054386785
    Avarage f1 score: 0.8549253034547153

Gaussian Naïve Bayes (ternary feature):
    Avarage precision score: 0.9109024309024308
    Avarage recall score: 0.9291280972417016
    Avarage f1 score: 0.9190921465979258

Bernoulli Naïve Bayes (impute missing values):
    Avarage precision score: 0.8243655675986503
    Avarage recall score: 0.911350319463924
    Avarage f1 score: 0.8641173038765781
```

From the results of Naïve Bayes, when training the dataset with ternary features, the Gaussian Naïve Bayes model has better scores on precision, recall, and f1, than the others. I think it might be because it does have a partial continuous property that the attribute we don't know can be seen as in the middle of yes or no, which is "0.5" in between 1 and 0.

Ultimately, for the dataset that instances with missing values are discarded and that missing values are imputed, I would choose the decision tree as the classifier due to the higher scores on precision, recall, and f1. In particular, the 'Entropy' criterion is preferred as I mentioned above. For the dataset with ternary features, I would choose Gaussian Naïve Bayes as classifiers since it presents decent scores on all precision, recall, and f1, which performs better than the decision tree.