# Introduction

The task is to build classification models to recognize the handwriting dataset. A good candidate should be able to take advantage of appropriate information from training set and successfully predict the majority of testing set.

The dataset has 785 attributes. The “label” column is the target variables, pixel sequences are the features(explanatory variables).

Imported by Pandas module, all attributes are identified as numerical variables. There is no missing value in the original dataset. The data type is integer. Hence the primary preprocessing is done. In the following sections, different methods require various additional preprocessing: normalization, discretization, etc.

To construct the classification models, this report spitted the training dataset into train data and test data. The test size is 30%, and the random state is 1, across all sections in this report.

Decision tree classification mapping features to the target variable. Each node in the tree represents a target value. Paths from the root to the final nodes combines conditions to obtain that value. Branches states for the value conditions of leading to each successor node from the present node. There are many ways to determine the conditions: equal frequency, equal width, entropy, etc.

Naïve Bayes access the predicted target value by calculating how likely the value is a specific category, given a specific sequence of feature outcome. It assumes all features are independent. The following is how to calculate the conditional probability:

$$P(target\ var = a \big| \{feature_i\ vaule\}_{i=0}^{783})$$

$$= \frac{P(target\ var = a) \times P(\{feature_i\ vaule\}_{i=0}^{783} | target\ var = a)}{P(\{feature_i\ vaule\}_{i=0}^{783})}$$

## Decision Tree Classifier

The decision-tree model relies on the primary settings. To balance the accuracy and overfitting issues, this report tried max depth of the tree by first. The list is 3,5,7,8,9,10. When max depth is achieved 8, 9 or 10, the accuracy scores of test data and train data has significant differences(greater than 2%), which leads to potential overfitting issue. Therefore, the range of max depth is set from 3 to 7.

Table-1 comparison between different settings

| Max depth | 10 | 9 | 8 | 7 |
|---|---|---|---|---|
| Accuracy of training data | 0.9311 | 0.8972 | 0.8595 | **0.8116** |
| Accuracy of test data | 0.8573 | 0.8481 | 0.8310 | **0.7928** |

The minimum split of each node is also non-trivial. In Orange, the default setting is 2. Thus this report set the range of experiment is from 2 to 5.

To explore optimal settings of the decision tree classifier, this report implied Grid Search and CV approach. The final results are when the max depth is 7, the minimum split is 3, and criterion='entropy'. Below is the result of the optimal setting.

Table-2 classification reports for the optimal decision tree model

| | **precision** | **recall** | **f1-score** | **support** |
|---|---|---|---|---|
| **0.0** | **0.90** | **0.91** | **0.90** | **1247** |
| **1.0** | **0.91** | **0.91** | **0.91** | **1395** |
| **2.0** | **0.76** | **0.79** | **0.77** | **1257** |
| **3.0** | **0.74** | **0.70** | **0.72** | **1321** |

| | | | | |
|---|---|---|---|---|
| **4.0** | **0.73** | **0.79** | **0.76** | **1232** |
| **5.0** | **0.73** | **0.69** | **0.71** | **1111** |
| **6.0** | **0.81** | **0.83** | **0.82** | **1258** |
| **7.0** | **0.85** | **0.82** | **0.84** | **1285** |
| **8.0** | **0.76** | **0.74** | **0.75** | **1249** |
| **9.0** | **0.73** | **0.73** | **0.73** | **1245** |

| | | | | |
|---|---|---|---|---|
| **accuracy** | | | **0.79** | **12600** |
| **macro avg** | **0.79** | **0.79** | **0.79** | **12600** |
| **weighted avg** | **0.79** | **0.79** | **0.79** | **12600** |

In over half of all categories, the decision tree has robust F1 scores. The overall assessment reflects how the model performs on most categories. The 3-fold cross validation is as follows:

**0.7909, 0.7972, 0.8055**

The accuracies over three folds are predictable, and the difference is insignificant. The primary conclusion is that the optimal decision tree does not have overfitting nor underfitting issue.

## Naïve Bayes

Naïve Bayes has different formats given the variables of features. First consider all attributes as continuous variables. Gaussian Naïve Bayes is the solution to estimate the probability density.

Before running into the model, additional data preprocessing is needed. Data is converted from integer to float. The next step is called feature scaling. It allows test data and train data on the same scale, though the difference here is not obvious.

Now it is safe to train the model. The accuracy score on test data is **0.5483,** and the score on train data is **0.5544.** The overfitting issue is not the major problem, but the level of accuracy is.

Table-2 classification reports for Gaussian Naïve Bayes

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| **0.0** | **0.64** | **0.92** | **0.76** | **1247** |
| **1.0** | **0.77** | **0.96** | **0.85** | **1395** |
| **2.0** | **0.85** | **0.24** | **0.37** | **1257** |
| **3.0** | **0.69** | **0.34** | **0.45** | **1321** |
| **4.0** | **0.82** | **0.16** | **0.26** | **1232** |
| **5.0** | **0.70** | **0.04** | **0.07** | **1111** |
| **6.0** | **0.63** | **0.94** | **0.75** | **1258** |
| **7.0** | **0.88** | **0.29** | **0.44** | **1285** |
| **8.0** | **0.31** | **0.58** | **0.41** | **1249** |
| **9.0** | **0.37** | **0.93** | **0.53** | **1245** |
| **accuracy** | | | **0.55** | **12600** |
| **macro avg** | **0.66** | **0.54** | **0.49** | **12600** |
| **weighted avg** | **0.67** | **0.55** | **0.50** | **12600** |

Gaussian Naïve Bayes performance indicates that it does not include all information and the prediction is not precise. Although it has relative precise prediction on 3 categories, in most cases, it lacks to the ability to capture the pattern. In some category, the F1-score is only 0.26. As this model didn't set any parameters, it might be helpful to switch to other Naïve Bayes approaches.

The multinomial distribution is a more generalized version of the binomial distribution. For $n$ independent experiments, each of which is realized to a success for exactly one of $k$ outcomes(features). The multinomial distribution shows the probability of a specific, random combination of numbers of successes for all kinds of outcomes.

First experiment Multinomial Naïve Bayes with only normalization. The data preprocessing is still needed. The training features are scaled by the Min-Max scaler, which makes sure that the value range is from 0 to 1.

Next dig into the model. The accuracy score on test data is **0.8263**, and the score on training data is **0.8246**. The overfitting issue does not exist. At the same time the prediction improved. Below is the detail report for this model:

Table-3 classification reports for Multinomial Naïve Bayes

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| **0.0** | **0.92** | **0.92** | **0.92** | **1247** |
| **1.0** | **0.88** | **0.94** | **0.91** | **1395** |
| **2.0** | **0.88** | **0.84** | **0.86** | **1257** |
| **3.0** | **0.80** | **0.80** | **0.80** | **1321** |
| **4.0** | **0.83** | **0.73** | **0.78** | **1232** |
| **5.0** | **0.84** | **0.66** | **0.74** | **1111** |
| **6.0** | **0.89** | **0.92** | **0.90** | **1258** |
| **7.0** | **0.94** | **0.83** | **0.88** | **1285** |
| **8.0** | **0.66** | **0.78** | **0.71** | **1249** |
| **9.0** | **0.68** | **0.82** | **0.74** | **1245** |
| **accuracy** | | | **0.83** | **12600** |
| **macro avg** | **0.83** | **0.82** | **0.82** | **12600** |
| **weighted avg** | **0.83** | **0.83** | **0.83** | **12600** |

The classification report describes that in most categories, the model fits well in most training and test data. However, in category 8 and 9, the precision has relatively downwards, which implies these predictions need improvement.

Next modify the data as discretization. The preprocess used 5 bins for every feature, which parallel to the default settings in Orange.

The accuracy score on test data is **0.8301**, and the score on training data is **0.8229**. The overfitting issue does not exist. At the same time the prediction on test data improved 0.4%. Below is the detail report for this model:

Table-4 classification reports for Multinomial Naïve Bayes(discretized features)

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| **0.0** | **0.92** | **0.91** | **0.91** | **1247** |
| **1.0** | **0.85** | **0.95** | **0.90** | **1395** |
| **2.0** | **0.82** | **0.81** | **0.81** | **1257** |
| **3.0** | **0.78** | **0.77** | **0.78** | **1321** |
| **4.0** | **0.81** | **0.83** | **0.82** | **1232** |
| **5.0** | **0.75** | **0.72** | **0.73** | **1111** |
| **6.0** | **0.89** | **0.89** | **0.89** | **1258** |
| **7.0** | **0.91** | **0.85** | **0.88** | **1285** |
| **8.0** | **0.82** | **0.77** | **0.80** | **1249** |
| **9.0** | **0.75** | **0.79** | **0.77** | **1245** |
| **accuracy** | | | **0.83** | **12600** |
| **macro avg** | **0.83** | **0.83** | **0.83** | **12600** |
| **weighted avg** | **0.83** | **0.83** | **0.83** | **12600** |

The classification report describes that the model fits well in the training and test data over categories. The primary conclusion is that the Naïve Bayes model does not have overfitting nor underfitting issue.

The multinomial Naïve Bayes model with discretized features has the relative optimal performance. The Gaussian Naïve Bayes has the lowest accuracy on test data, leading to misclassification. The multinomial Naïve Bayes improves the accuracy by more than 20%. Follow-up work on discretization enhance the precise of prediction further, by

0.4%. It is not a very promising change, given the earlier change on switching methods. However, considering the large scale of dataset and the test dataset, this improvement could reduce hundreds or thousands of misclassifications in the future. Therefore, the report recommends applying multinomial Naïve Bayes(discretized features) in the future test.

## Comparison algorithms

The two algorithms in this report have the following results:

Table-5 performance for each method

| | DTC | GNB | MNB | MNB-discretization |
|---|---|---|---|---|
| Time(second) | 34.00 | 13.76 | **7.42** | **8.97** |
| Test data | 0.7928 | 0.5483 | 0.8263 | **0.8301** |
| Training data | 0.8116 | 0.5544 | 0.8246 | **0.8229** |

Note: the time includes data preprocessing, training model, prediction test data, and reporting performance.

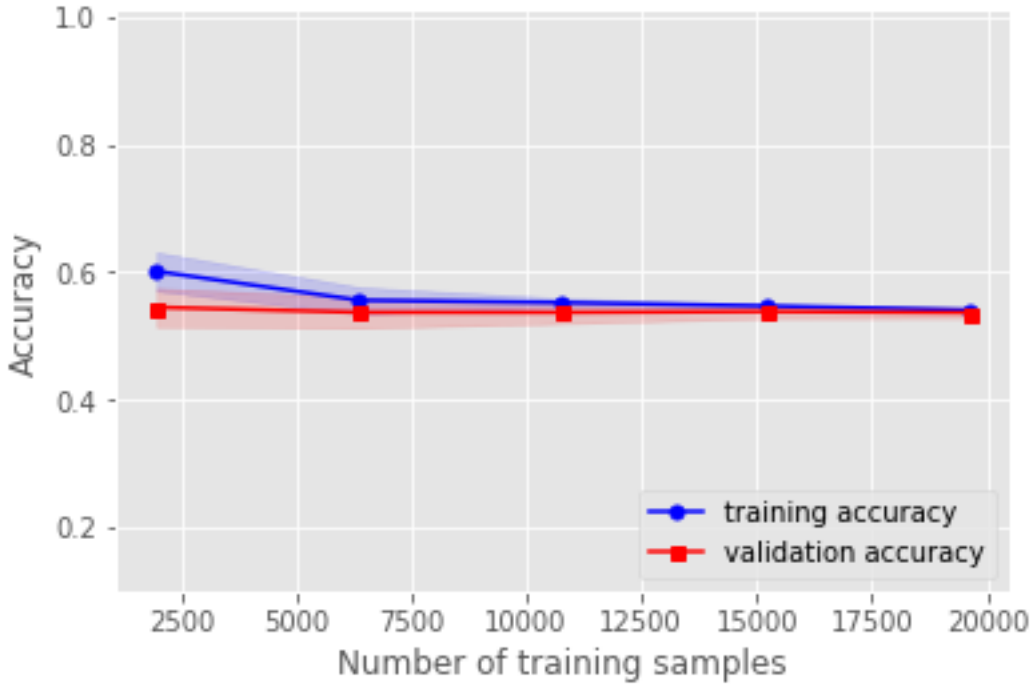


Figure-1 learning curve of Gaussian Naïve Bayes

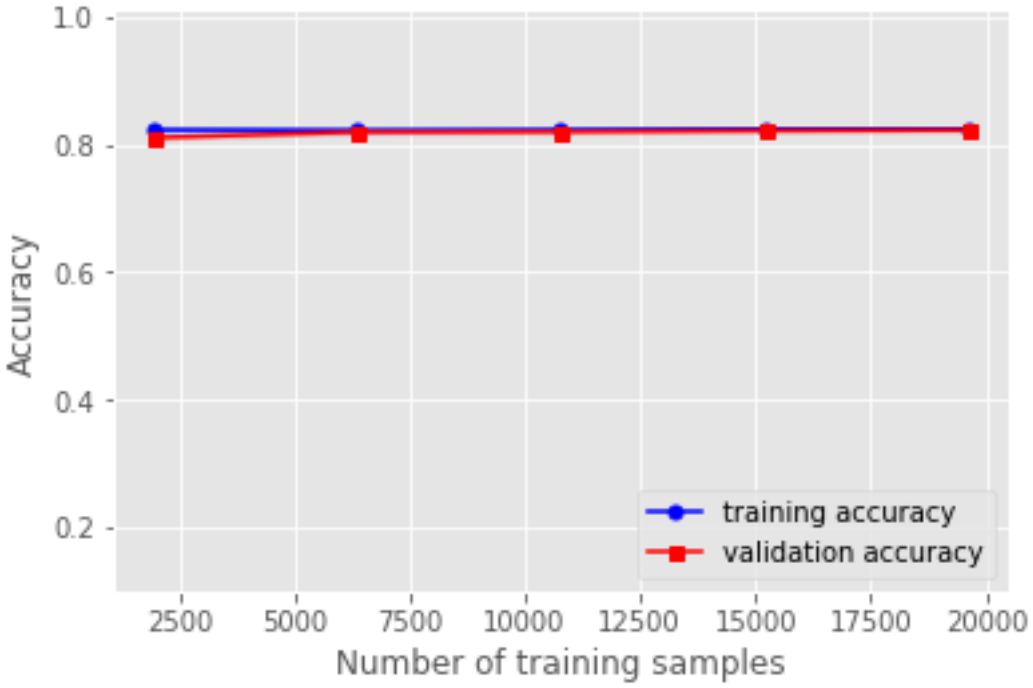


Figure-3 learning curve of Multinomial Naïve Bayes

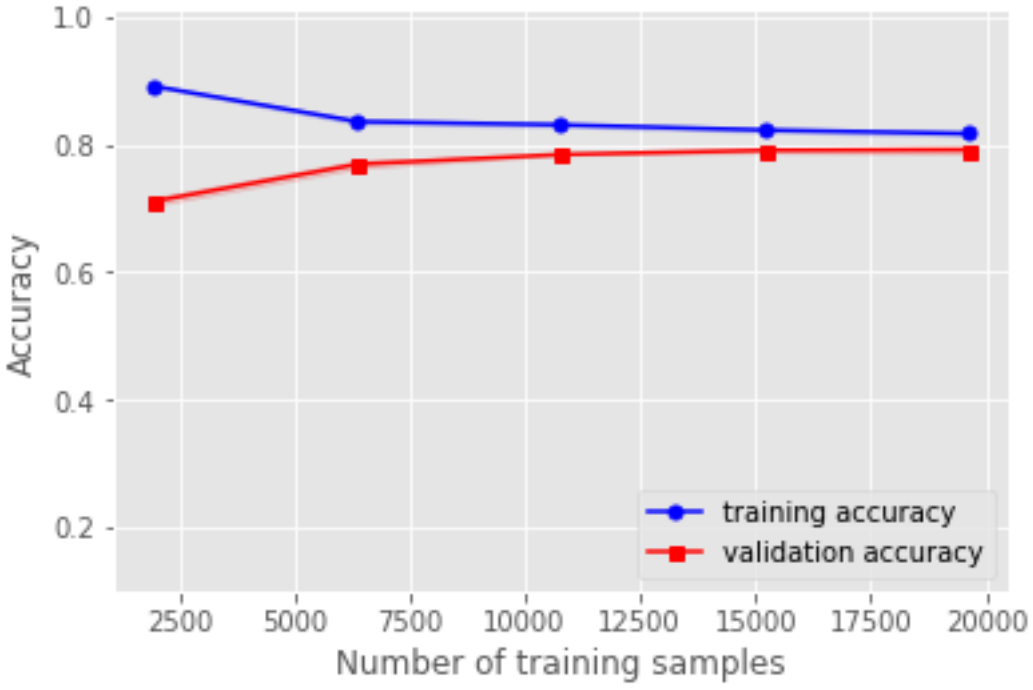


Figure-3 learning curve of Decision Tree

*The decision tree has the second-best accuracy score*. However, *decision tree is time-consuming*. It spent twice time of GNB's, and four time of MNB's. The reason behind

the above results is how to construct the tree. It exhausts every possible combination to finalize each node with different branches. Then, the paths lead to each category includes all essential information to determine the observation belongs to this category. With a large scale of dataset and multiple values in each feature, the calculation in training process costs a lot of time. Decision tree is straightforward and insensitive to extreme values. With the visualized tree map, it is accessible to classify test dataset in the future.

*The multinomial Naïve Bayes with discretization has the best accuracy score and spent the second least of time. The Gaussian Naïve Bayes has the lowest accuracy score and spent the third least of time.* Unlike GNB, MNB simplified features into five values, which is easy to calculate and saved time. The accuracy difference between MNB and GNB rises from discretization. The value of each feature varies from zero to hundreds, and the value of target variable is limited. If features are determined as continuous variables, the probability of specific feature combination appears given some value of target variable is not highly precise, compared to MNB. If features are determined as discrete variables, the reduction of value range for features largely rose up the precision of how likely the feature combination appears, given the specific value of target variable. As for more detail explanation, we need more information of each feature.

From the above graphs, selected models avoided overfitting issue. The learning curve shows that both accuracy of test and train data converge as the sample size blows up. Hence those accuracy and prediction results is reliable.

In conclusion, if the feature represents different status of handwriting, then discretization makes sense. Both Multinomial Naïve Bayes and Decision tree are good candidates. If features measure specific values from continuous functions of handwriting, MNB should be removed from the list.