Data Mining HW4

Scikit-Learn

Name: 劉廷緯 Department: 電信一 Student ID: R07942089

1. **News Dataset: Testing label is provided**
   a. **Implement Naive Bayes on News dataset**
      i. **What's the parameters and performance of your best model ? (Baseline: Test accuracy 85%) [10%]**

      The parameter I tune for Naive Bayes is the **alpha** value:
      naive_bayes.MultinomialNB(alpha=0.065) => 0.89511

      ii. **Compare different distribution assumption, which is the most suitable for News dataset ? List the testing accuracy. [5%]**

```
>> [Naive Bayes Runner] Guassian - Accuracy: 0.8097902097902098
>> [Naive Bayes Runner] Multinominal - Accuracy: 0.8951048951048951
>> [Naive Bayes Runner] Complement - Accuracy: 0.8881118881118881
>> [Naive Bayes Runner] Bernoulli - Accuracy: 0.8272727272727273
```

- naive_bayes.GaussianNB() => 0.80979
- **naive_bayes.MultinomialNB(alpha=0.065) => 0.89511**
- naive_bayes.ComplementNB(alpha=0.136) => 0.88811
- naive_bayes.BernoulliNB(alpha=0.002) => 0.82727

The most suitable distribution assumption for the News dataset is **Multinominal**, all distribution is compared with its best parameter. (Will describe how to find best parameter in the question c.)

   b. **Implement Decision Tree on News dataset**
      i. **What's the parameters and performance of your best model ? (Baseline: Test accuracy 61%) [10%]**

      The parameter I tune for Decision Tree is the **max_depth** value, furthurmore I use 'random' as splitting method:
      tree.DecisionTreeClassifier(criterion='gini',
      splitter='random',
      random_state=1337,
      max_depth=64) => 0.64895

```
>> [Decision Tree Runner] - Accuracy: 0.6489510489510489
```

c. **How do you choose the parameters to get the best model ? [5%]**

As mentioned above, I tune the **alpha** value for Naive Bayes, and tune the **max_depth** value for Decision Trees. In both cases, I brute search different parameter values in a given range, and evaluate the model's performance under that parameter value with the testing set. The value that results in the highest testing accuracy is then chosen as the best parameter for that model.

For alpha, these values are searched for: **np.arange(0.0001, 1.0, 0.0001)**
For max_depth, these values are searched for: **np.arange(1, 64)**

2. **Mushroom Dataset: Testing label is provided**
   a. **How do you preprocess the mushroom dataset? [5%]**

In the preprocessing stage, simple one-hot encoding is utilized, in which 22 categorical attributes are transformed into a 117 dimension one-hot vector, resulting data shape are shown:

```
>> [Data Loader] Reading the Mushroom dataset...
>> [Data Loader] Training x data: (6500, 117)
>> [Data Loader] Training y data: (6500,)
>> [Data Loader] Testing x data: (1624, 117)
>> [Data Loader] Testing y data: (1624,)
```

b. I**mplement Naive Bayes on mushroom dataset**
   i. **What's the parameters and performance of your best model ? (Baseline: Test accuracy 98%) [10%]**

The parameter I tune for Naive Bayes is the **alpha** value:
naive_bayes.MultinomialNB(alpha=0.0001) => 0.99569

   ii. **Compare different distribution assumption, which is the most suitable for mushroom dataset ? List the testing accuracy. [5%]**

```
>> [Naive Bayes Runner] Guassian - Accuracy: 0.9550492610837439
>> [Naive Bayes Runner] Multinominal - Accuracy: 0.9938423645320197
>> [Naive Bayes Runner] Complement - Accuracy: 0.9932266009852216
>> [Naive Bayes Runner] Bernoulli - Accuracy: 0.9870689655172413
```

- naive_bayes.GaussianNB() => 0.95505
- **naive_bayes.MultinomialNB(alpha=0.0001) => 0.99569**
- naive_bayes.ComplementNB(alpha=0.0001) => 0.99507
- naive_bayes.BernoulliNB(alpha=0.0001) => 0.98830

The most suitable distribution assumption for the News dataset is **Multinominal**, all distribution is compared with its best parameter.

c. Implement Decision Tree on mushroom dataset

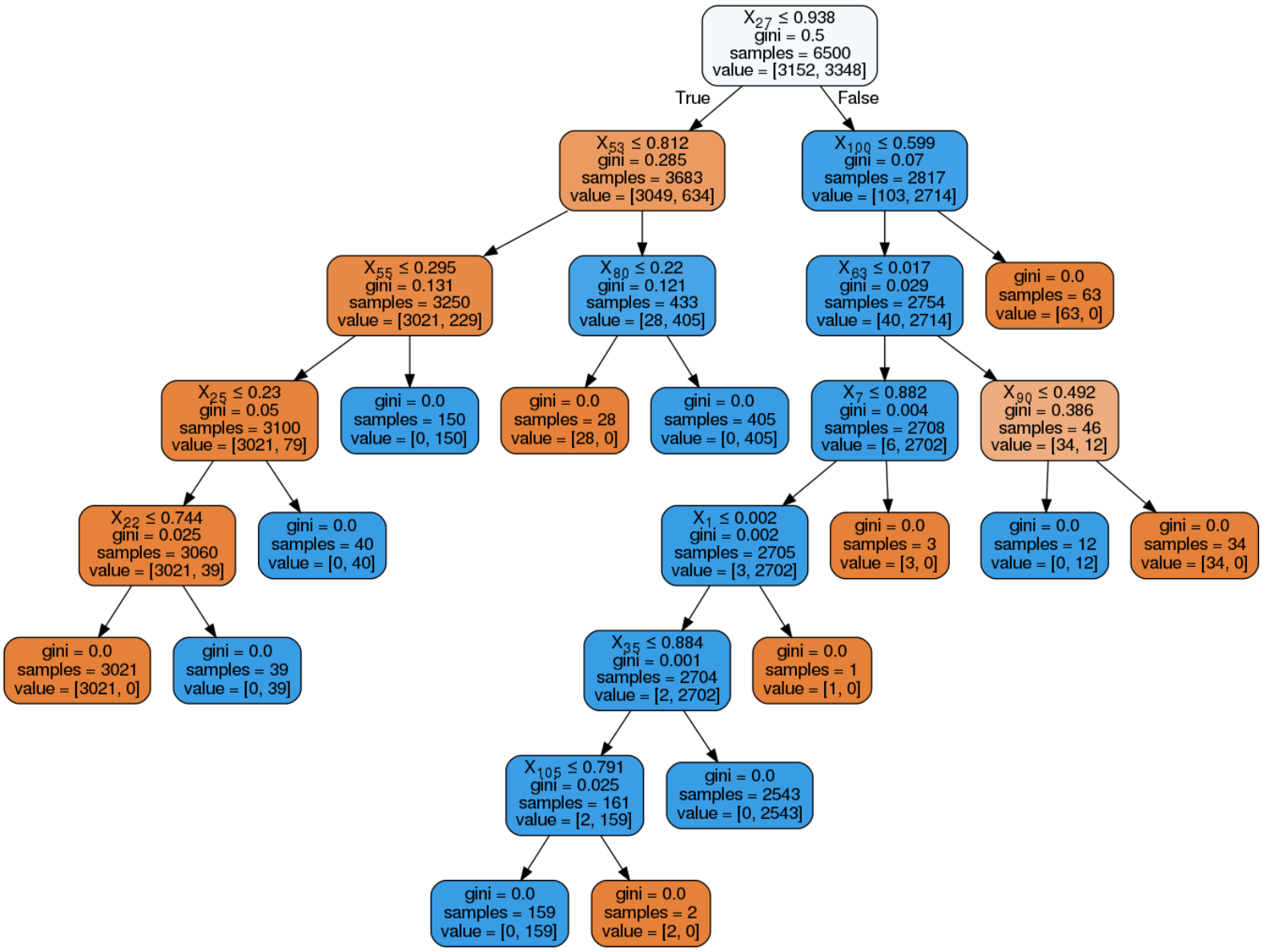
i. **What's the performance of your best model ? (Baseline: Test accuracy 99%) [10%]**

The parameter I tune for Decision Tree is the max_depth value, furthurmore I use 'random' as splitting method:

tree.DecisionTreeClassifier(criterion='gini',
splitter='random',
random_state=1337,
max_depth=64) => 1.0

```
>> [Decision Tree Runner] - Accuracy: 1.0
```

ii. **Use graphviz tool to plot your decision tree [5%]**



d. **Observe the data properties of News and mushroom dataset. According to the model performance, what kind of dataset is more suitable for naive bayes / decision tree ? [5%]**

According to model performances on the News and mushroom dataset, we can easily observe that the **mushroom dataset is more suitable** for both naive bayes and decision tree classifiers. In the News dataset, each sample is a 23909 dimension TF-IDF feature vector, which is clearly too large and too spare for the classifiers to learn. The mushroom dataset, on the other hand, each sample only has a 117 dimension one-hot feature vector after my preprocessing. Hence the kind of dataset that both naive bayes and decision tree classifiers can perform well are those that has **feature vector size roughly in the hundredth magnitude**.

3. **Income Dataset: Testing label is not provided**
   Implement Naive Bayes and Decision Tree on income dataset
   a. **How do you preprocess the data ? Missing value ? [10%]**

The data preprocessing pipeline is as follows:

- Specify each entry to either one of the data type: **(int, str)**
- Identify all missing entries **'?'** and replace them with **np.nan**
- **Impute and estimate all missing entries**:
    - if dtype is **int**: impute with mean value of the feature column
    - If dtype is **str**: impute with most frequent item in the feature column
- **Split data into categorical and continuous** and process them separately:
    - **categorical** features index = [1, 3, 5, 6, 7, 8, 9, 13]
    - **continuous** features index = [0, 2, 4, 10, 11, 12]
- **For categorical data:** 8 categorical attributes are transformed into 99 dimension **one-hot** feature vector
- **For continuous data: Normalize** with maximum norm of that feature column
- **Re-concatenate** categorical features and continues features, the resulting data shape is shown:

```
>> [Data Loader] Reading the Income dataset...
>> [Data Loader] Training x data shape: (32562, 105)
>> [Data Loader] Training y data shape: (32562,)
>> [Data Loader] Testing x data shape: (16280, 105)
```

b. **Which model gets better performance ? Show the parameters. (Surpass the weak baseline (Test accuracy: 80%) for 10%. Strong baseline (Test accuracy: 85%) for 10%)**

The **decision tree** model performed better on this dataset, the accuracy of both naive bayes models and decision tree model are shown, all evaluate with N-fold

cross-validation with N=10. The best parameter for these models are found in the same way as in 1-c, however I use N-fold cross validation to evaluate the parameter values instead of the testing set accuracy.

```
>> [Naive Bayes Runner] Guassian - Accuracy: 0.5860226359726692
>> [Naive Bayes Runner] Multinominal - Accuracy: 0.7914756157842513
>> [Naive Bayes Runner] Complement - Accuracy: 0.7499246241807673
>> [Naive Bayes Runner] Bernoulli - Accuracy: 0.7576022387121311
```

Naive Bayes on Income dataset:

```
>> [Decision Tree Runner] - Accuracy: 0.8603584507723732
```

Decision Tree on Income dataset:

Note that these are **average N-fold cross validation accuracies**, and not the testing accuracy. We see that the **Decision Tree model performs better**, all the parameters are listed as follow:

- naive_bayes.GaussianNB() => 0.58602 (baseline)
- **naive_bayes.MultinomialNB(alpha=0.959) => 0.79148**
- naive_bayes.ComplementNB(alpha=0.16) => 0.74992
- naive_bayes.BernoulliNB(alpha=0.001) => 0.75760

- **tree.DecisionTreeClassifier(criterion='entropy',**
  random_state=1337,
  min_impurity_decrease=2e-4
  max_depth=15) => 0.86036

We use the **entropy** criterion here instead of gini, and set an additional **minimum impurity decrease** threshold, different from the other decision tree settings in the previous sections.