2102 model doc

Manual:

```
[1]: import pandas as pd
     import numpy as np
     import matplotlib.pyplot as plt
     from sklearn.model_selection import train_test_split
     from sklearn.pipeline import Pipeline
     from sklearn.linear_model import LogisticRegression
     from sklearn.tree import DecisionTreeClassifier, plot_tree
     from sklearn.model_selection import GridSearchCV, GroupKFold, RandomizedSearchCV
     from sklearn.metrics import confusion_matrix, ConfusionMatrixDisplay, f1_score, classifi
     from sklearn.utils import shuffle
     import warnings

     warnings.filterwarnings("ignore")

     # 定义逻辑回归模型和决策树模型
     logistic_model = LogisticRegression(multi_class='multinomial',solver='lbfgs')
     tree_model = DecisionTreeClassifier(criterion='entropy',random_state=42)

     # 定义Pipeline对象
     pipe_logistic = Pipeline([('model', logistic_model)])
     pipe_tree = Pipeline([('model', tree_model)])

     # 定义超参数搜索空间(because our dataset still have noise after denoise)
     param_grid_logistic = {'model__C': [0.001, 0.01, 0.1, 1, 10, 100]}

     param_grid_tree = {'model__max_depth': [3, 5, 7, 10, 15, 20, None],
                        'model__min_samples_split': [2, 5, 10, 20, 30, 40, 50],
                        'model__min_samples_leaf': [1, 2, 5, 10, 15, 20]}

     # choose to drop which columns
     drop_list = ['filename', 'label', 'band_energy_ratio_800', 'band_energy_ratio_1600', 'du
```

The highlighted part can control the features included in the model; the filename and label must be included, and other parts are the feature you want to remove which is hard to explain (if you want to include mfcc s in the model, please remove them from the list)

Result: remove the mfcc will affect the accuracy (f1-score for classification problem) of logistic regression but not for the tree model (in the above train and test predict accuracy)

Keyword:

F1-score: a score that can describe the confusion matrix well, and the confusion matrix is the criteria to evaluate the accuracy.

Explanation:

In the model part, we aims to build a multinormal logistic regression and tree model for identify the class of the sub-audio and recognize the class of original audio by the predicted result of the sub-audio which separated from the original recording.

How to set the train set and the test set, and the k-fold cross validation (by group and stratified k-fold)

(we use the fifth fold as test set when split train and test set)

group k-fold

Reason: The samples with the same filename are the sub-audio extracted from the same original audio file.

(If some sub-audios with a particular filename go to the train set and some go to the test set, it would be a problem because the model may recognize the class of sub-audios by the pattern that appeared in the particular original audio with that filename rather than by the pattern that the audio with the same label always appeared.) (Optional--just explained to you)

It will cause the data leakage problem and overestimate the accuracy, which is the f1-score for the classification problem when the samples with the same filename are split into different sets (i.e., train set and test set). The situation may occur even if the feature does not include the filename. This is similar to the patient overlap problem in the medical image recognition task.

Therefore, we group the sub-audio with the same filename in the same fold to ensure it will not split into different sets.

Stratified k-fold

Reason: To ensure there would not be an extreme situation where there are just a few samples for a label in the train set, making the model not fit the data well for that label.

By stratifying the data, we can ensure that each fold has a similar proportion of each label (class), which can improve the overall performance and reliability of the model.

Grid Search and Hyperparameter

To prevent the overfitting problem for the model, we need to tune the hyperparameter to limit the complexity of the model. For example, we set the hyperparameter for the tree model to do the pre-pruning process.

We search for the best hyperparameter and conduct the best model for the audio classification problem by trying every combination of the hyperparameter list we set and using the f1-score as the evaluation criteria.

We use cross-validation in Grid Search with group and stratified k-fold to set the fold which mentioned before

Report of the model

Multinormal logistic regression:

```
[ ]: #get the report of logistic regression
     #get the model from pipeline
     model = best_logistic.named_steps['model']

     # 创建 DataFrame 对象
     coef_df = pd.DataFrame(model.coef_, columns=X_train.columns, index=model.classes_)
     intercept_df = pd.DataFrame(model.intercept_, columns=['Intercept'], index=model.class

     # 添加列名，表示每个类别
     coef_df.columns.name = 'Class'

     # 将 intercept_df 合并到 coef_df 中
     result_df = pd.concat([intercept_df, coef_df], axis=1)

     result_df.T
```

[5]:

	cat	cow	dog
Intercept	-0.005537	0.003339	0.002198
AUC	-0.000126	0.000653	-0.000527
max_freq	0.000617	-0.002395	0.001778
spectral_centroid	0.002801	-0.002087	-0.000715
one_ratio_band_energy_freq	0.001810	0.000524	-0.002334
power_bandwidth	-0.000606	0.000223	0.000383
kurtosis	0.002864	0.002485	-0.005349
skewness	-0.000177	0.000607	-0.000430
zcr	0.000011	0.000047	-0.000059
spectral_entropy	-0.002891	0.001589	0.001301
spectral_skewness	-0.029235	0.016092	0.013143
spectral_spread	-0.003667	0.002951	0.000717
fundamental_frequency	-0.000848	-0.001894	0.002742
interquartile_range	-0.002034	-0.000327	0.002362
turning_rate	0.000322	-0.000150	-0.000172

Intercept: when all the feature is 0 (may represent that the audio have no sound), what would be the probability of that audio belonging to each class (cat, cow, dog)
Other feature: When the feature increases one unit, how would it affect the probability of that audio belonging to each class (please refer to the lecture slide)
(p.s. You can remove the feature that you think is hard to explain in the drop list in first cell; refer to the part in the opening in this doc)

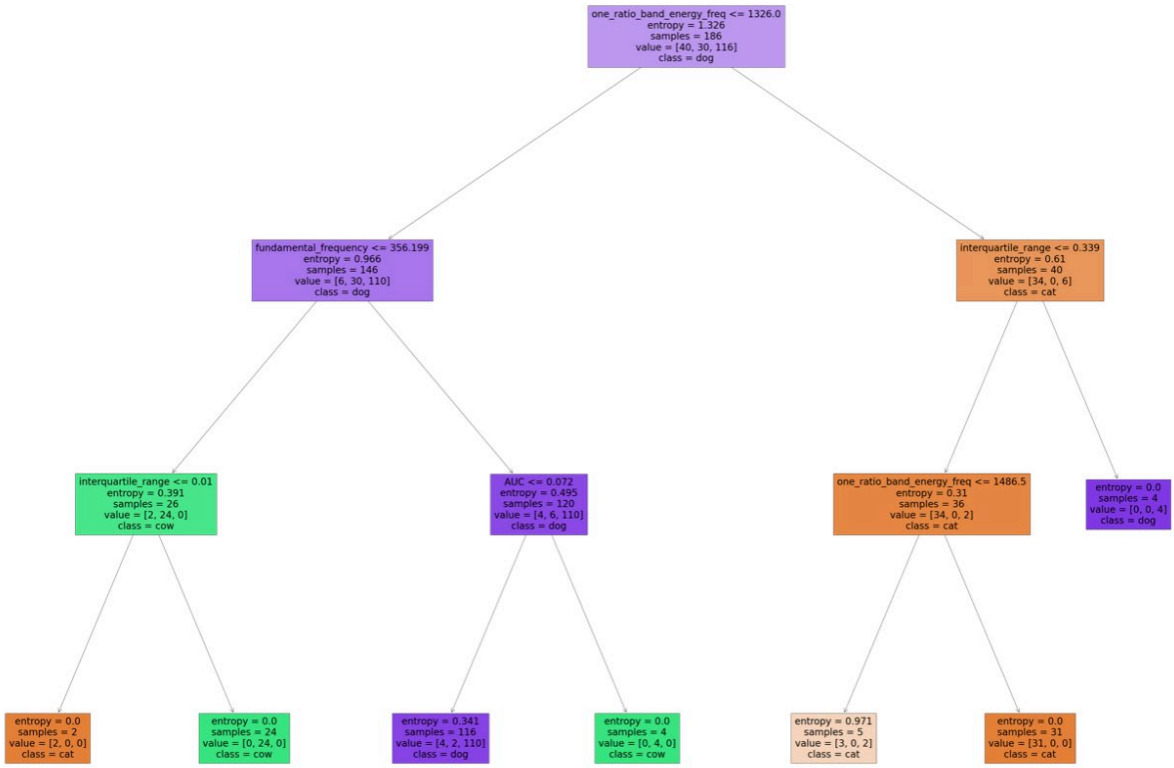
Tree (for each node):
https://stackoverflow.com/questions/23557545/how-to-explain-the-decision-tree-from-scikit-learn
Line1: the split criteria →true go left, false go right.

Line 2: impurity→refer to lecture slide.

Line3: the sample goes through this node

Line4: the number of samples of each class goes through this node (cat, cow, dog)

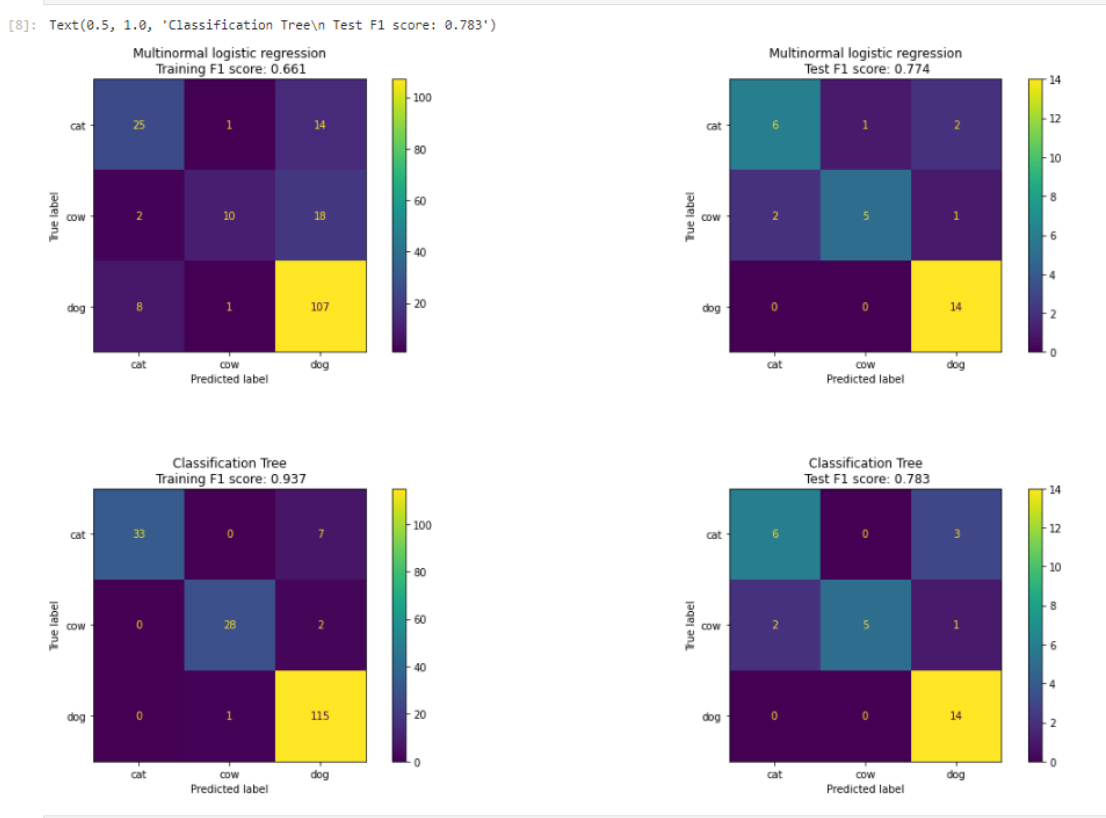
Line5: if it is the leaf node, it means that if the sample goes to this node, what will the model predict as its class. (just the class that the most sample belong to in this node)



The performance for the model:

(The graph below is not include mfcc, if you have any change in the set of feature included the model, you should plot it again)

(p.s. Notes that the plot measure the accuracy in both using training to classify training set and test set. Including mfcc or not seems affect much in training set predict but this statement is not accuracy because it can be a special situation which we choose the fifth fold as test set, but the fact is that removing mfcc does affect the performance of logistic model.)



Using the prediction result to classify the class of original audio:

In our expectation, if the sub-audio is more important for original audio, it would be have a large sound compared with other sub-audio (just like if we aims to record the sound of cow, we will put the mic to it more closely, more smaller sound is more likely to be noise. For example, a sound of a pig which is the neighbor of the cow that we like to record its sound.

Therefore, we choose AUC (which can show the loudness of the audio), calculate weighting(the importance) by this. Finally, we add the weighting of each class and calculate the probability of each class (don't know how to say, the process like: (class of sub-audio: weighting) cow:0.7, dog:0.2, cow:0.1 →Prob of original audio:

cow: 0.7+0.1=0.8 dog: 0.2

So that audio is cow)