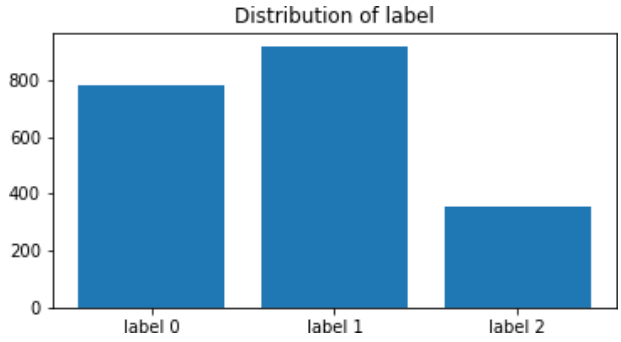
Name: Po Kit Man

Data Exploration

The first step is data exploration. We first check the distribution of labels 0, 1 and 2. From the histogram below, we can see that the data is imbalanced. The numbers of label 0 and label 1 are at least two times of that of label 2.



Correlations between features

From the code, we can see that approximately 25% of features with high correlation coefficient of over 0.9. Since there are a large numbers of features with high correlation with each other, we will perform PCA later to reduce the dimensionality.

Data Cleaning

Check and remove outliers

There are 2056 records with outliers. Since the total number of records is 2060 and the number of outliers is almost 99.8% of all the records, we will not remove records with outliers.

Check the records with missing values

There are no records with missing values.

Duplicated rows and columns

There are 194 duplicated rows. Since each row is still a new sample and having duplicated rows just mean that the same features are extracted to represent the same sample. Duplicated rows are beneficial because it can provide more support to the splits since multiple instances can confirm the split is directly related to the classification. As a result, duplicated rows will not be removed. For the duplicated columns, the duplicated columns will be removed after performing PCA.

Feature Selection (Perform PCA)

Since there are 2048 features. Some of them are highly correlated as mentioned above. Performing PCA can speed up the time for processing and improve the accuracy of the model. In this report, the proportion of explained variance is set to be 0.9. After performing PCA, there are 39 attributes left.

Decision Tree with standard parameters

After performing PCA, we will train the data with the decision tree model with the standard parameters first. For the decision tree model, we will evaluate the performance of the model by using Stratified K-Fold cross validation. We will use f1 weighted score as the metric for performance evaluation. Stratified K-Fold cross validation is the variation of K-Fold cross validation. In Stratified K-Fold cross validation, each fold is ensured to be an appropriate representative of the original data and each class is equally represented across each test fold. The percentages of each class in the entire data will be approximately the same for each test fold. Since the train data is imbalanced as mentioned above, using Stratified K-Fold cross validation can give a more unbiased and accurate result. F1 weighted score is the weighted average of f1 scores calculated for each label. It is used to measure the accuracy of the prediction of the model and f1 weighted score is a suitable measure for the classification problem with imbalanced data. In this case, we will use K=10. From the code, the average f1 weighted score is 0.764.

Hyperparameter tunning for Decision Tree model

There are two parameters to fine-tune in this model. They are “criterion” and “max_depth”. “criterion” is the function to measure the quality of a split. Supported criteria are “gini” for the Gini impurity and “entropy” for the information gain. “max_depth” is the maximum depth of the tree. In this hyperparameter tunning, the value range for “max_depth” is set between 1 and the number of features after PCA, which is 39, inclusively.

Then GridSearchCV is applied to test all the combinations of the hyperparameters and return the values we want. It will use Stratified 10-Fold cross-validation and f1 weighted score to evaluate all the possible combinations of hyperparameters values.

After Grid Search , best combination of hyperparameter values is found. For “criterion”, the value is “gini”. For “max_depth”, the value is 8. The corresponding f1 weighted score for the best combination of hyperparameters is 0.795.

Ensemble method: Random Forest with standard parameters

Then we will train with the random forest model. Random forest is an ensemble method for Decision Trees, generally trained via the bagging method. The concept behind random forest is the wisdom of crowds. Random forest includes a large number of individual decision trees. Each decision tree in the random forest gives out a class prediction and the class with the most votes will become the random forest model’s prediction.

We will first use the random forest models with the standard parameters first. Stratified 10-Fold cross validation and f1 weighted score will be applied to evaluate the performance of the model.

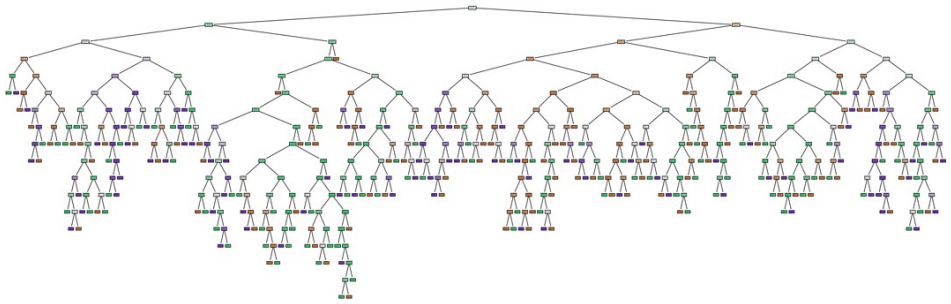
From the code, the average f1 weighted score is 0.907.

Hyperparameter tunning for Random Forest model

There are three hyperparameters to fine-tune in the random forest model. They are “criterion”, “max_depth” and “n_estimators”. The value range for “criterion” and “max_depth” are the same as that in the hyperparameter tunning for decision tree model. “n_estimators” is the number of trees in the forest. The default value of “n_estimators” is 100. We will use 200, 400, 600, 800 and 1000 for the value of “n_estimators”.

After applying GridSearchCV, the best combination of hyperparameter values is found. For “criterion”, the value is “gini”. For “max_depth”, the value is 37. For “n_estimators“, the value is 600. The corresponding f1 weighted score for the best combination of hyperparameters is 0.917.

Diagram of one of the decision tree in the Random Forest model



The Histograms of the Three Classes