Written by: Behnaz Nasiri October 17th 2020

Introduction:

In this project we try to use the provided data for a binary classification based on dependent feature “Type band” category for Cylinder Bands Data Set.
We can categorize the procedure to four steps as data preprocessing, feature selection, model building and the results.
We will choose Decision Tree to train and test our model.
The first step relates to the data analysis. In this step includes handling the missing values from our dataset and perform a data cleaning and use interpolation technique to estimate unknown variables.

1. Data Preprocessing:

First we open the band.data and read the data by splitting each line to 40 part. Using python library Pandas we put the data into a data frame. We can see that our data-set contain 550 row as instances and 40 attributes.

- **Missing values for numerical values:**

We need to get rid of the irrelevant instances with invaluable information. We delete two rows of the data frame which inly include ‘None’. Now we need to clean the data from those instances with too many missing data. Before removing instances with missing data we can find a relationship between missing values and Type band. We need to make sure not to remove if those instances has the effect in Type band. To avoid removing too many instances, first we find the instances contain missing data above the certain amount e.g. 13 data with ‘?’ And keep the others. After delete the instances our data-set shrink to 524 row.
Now to estimate the value for missing data we convert all the ‘?’ to null and then calculate the mean of each numeric column and round it to nearest integer for each column. (We will round to 2 decimal point for those column that all the values are not integers.)
Note that we should use the median for the ordinal categories with outliers.

- **Missing values for categorical values:**

First we find the categorical features which include missing values from our dataset. Then we find how many percentage of each our categories having missing values. To handle them my approach was to find the most frequent values in each category and replace the missing values with them. We can use replace method or interpolate method to fill the missing values.
We can also create a new label for missing data and call them “missing” in our data-set.

2. Feature Selection and removing rare labels:

Feature selection is a process in which we can identify and select the important features which are most relevant to our target variable. Feature selection can play a very important role in our process. It is important to find the features that are highly correlated to our target. By eliminating non correlated features we can clean our data space from redundant features. Therefore, we need to figure out which features to drop and still achieve a good result.
Now we split our data frame to two numeric and nominal category. Also in nominal groups we have ordinal and categorical.
Numerical variables are two types one is continuous variable and one is discrete variables.

- **Numerical variables:**

Numerical variables are two types one is continuous variable and one is discrete variables.

We can simply find the numeric variables with a line of code. It is important to find a relationship between independent variables and target variables. As can be seen from our data frame our numeric values are all continuous which are 20 . One way to visualize and explore the data is plotting them using Seaborn function. In this way we can understand how the variables are distributed and if there is any meaningful relationship between them. We can also visually capture the outliers.
Using scattered plot is a good way to see the effect of each category and remove the irrelevant feature from dataset. Figure (2.1) is a sample of pair-plot of three numeric category. This way we can find the correlation between our data.
As we can see in Figure (2.1), these categories have overlap in separating band type and may not play an important role in classifying band type.
Also we will drop one column name “ink colour” as all the variable was has same value. Not that we can not use this for our categorical variables.

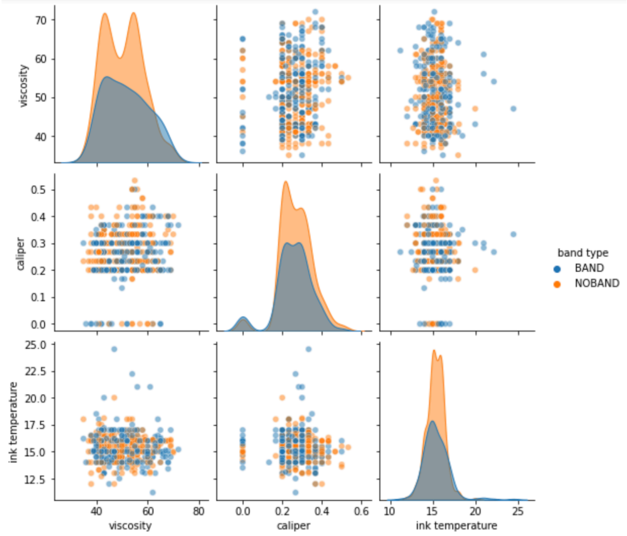

Figure (2.1): scattered plot for three numeric attributes

- **Categorical variables:**

We need convert all the nominal values to numeric values since the sklearn accept only float values. Before doing any encoding we need to make sure that we read all the nominal values in each category in a right way. For instance in some categories we have (YES and yes or NO and no), to avoid our model consider them as different class, we uppercase all the string.
One way to handle categorical variables them is ordinal encoding. It is related to the categories in which the values in nominal category have orders. For instance categories with yes and no values can be treated as binary (no: 0 and yes: 1) or “unit number” category can be encode to ordinal numbers. Ordinal encoding also can be used for other categories for example choosing numbers from 0 to 2 for “cylinder division” attribute (0: gallatin, 1: warsaw, 2: mattoon). Another way is called one-hot encoding. It maps each group of a categorical attribute to its own column. Then we will put 1 for in each row for the category it belongs to, and a zero otherwise. In this study we choose ordinal encoding for all the nominal features.

- **Visualize the relations between nominal and numeric features**

We can also find the relation between numeric attributes with categorical attributes with box plot. Figure (2.2) shows the box plot for “roller durometer” and “press type”.
Here we see that the variables in some categories are the same for all instances. In this case we drop those attributes as they make no difference in our classification. Like “ cylinder division” which is “gallatin” for all the instances.

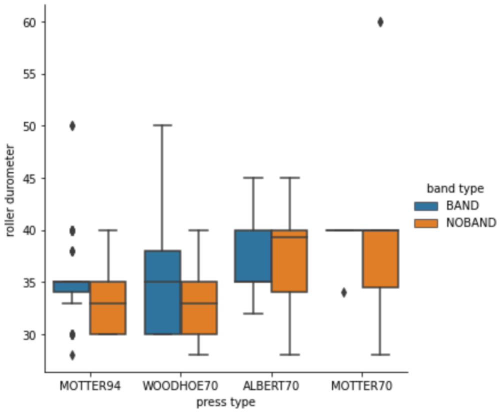

Figure 2.2: box plot for a categorical vs numeric attribute

In order to find the best categorical features for our model we can use two most common feature selection method called “chi-squared” and “mutual statistics”.
 We drop some identifier categories like “job number”, “‘customer” and “paper mill location” and “cylinder number”.

3. Model Builder

We use decision tree as a classifier for this data-set. Using Sklearn python library we split the data to train set and test set. It is randomly chosen with the ratio of 80 to 20 for train-set and test-set respectively. We will have 105 test sample and 419 train sample.

4. Results

The accuracy of model for the test set is around 76 % for decision tree classifier.
The ROC curve of the model has been shown in figure (4.1). Figure (4.2) also shows the Precision-Recall curve. The f1 score and AUC has been shown in this figure. We can see the generated results in table below.

Accuracy	Recall	Precision	F1 score
0.762	0.841	0.768	0.791

AUC	fpr	Tpr
0.706	0.381	0.841

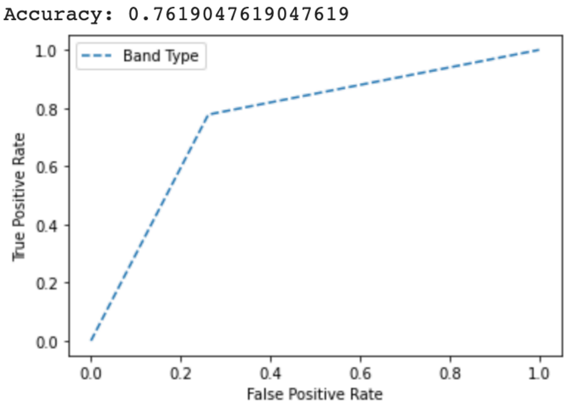

Figure (4.1): ROC curve

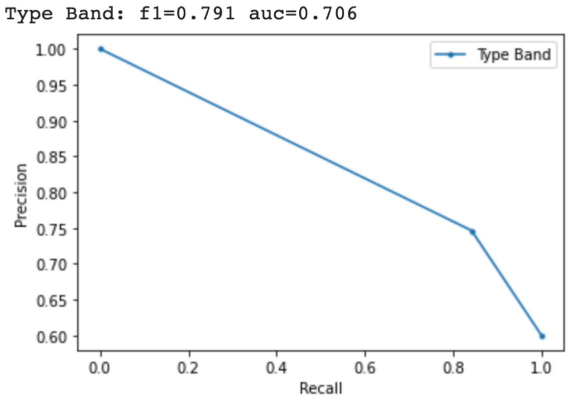

Figure (4.2) : AUC curve

- **Future work**

Due to tight time it is not possible to perform all the possible analysis on the data. For instance we can find the effect of missing data by creating another column and put the missing values as 1 and others as 0 and compare see its effect on our label data. This way we can understand how important it is to fill the missing values for each category.
We can also use oneHot Encoding for converting nominal features to numeric values.
Visualizing features also could be good way to find the informative features.
We can also use other classifier like SVM, Logistic Regression and etc, change the hyper parameters to find the more accurate result.