Curtin University – Department of Computing

Assignment Cover Sheet / Declaration of Originality

Complete this form if/as directed by your unit coordinator, lecturer or the assignment specification.

Last name:	Choppradit	Student ID:	20303349
Other name(s):	Pup (Pakcheera)		
Unit name:	Data Mining	Unit ID:	COMP3009
Lecturer / unit coordinator:	Dr. Sonny Pham	Tutor:	–
Date of submission:	8/10/21	Which assignment?	(Leave blank if the unit has only one assignment.)

I declare that:

- The above information is complete and accurate.
- The work I am submitting is *entirely my own*, except where clearly indicated otherwise and correctly referenced.
- I have taken (and will continue to take) all reasonable steps to ensure my work is *not accessible* to any other students who may gain unfair advantage from it.
- I have *not previously submitted* this work for any other unit, whether at Curtin University or elsewhere, or for prior attempts at this unit, except where clearly indicated otherwise.

I understand that:

- Plagiarism and collusion are dishonest, and unfair to all other students.
- Detection of plagiarism and collusion may be done manually or by using tools (such as Turnitin).
- If I plagiarise or collude, I risk failing the unit with a grade of ANN ("Result Annulled due to Academic Misconduct"), which will remain permanently on my academic record. I also risk termination from my course and other penalties.
- Even with correct referencing, my submission will only be marked according to what I have done myself, specifically for this assessment. I cannot re-use the work of others, or my own previously submitted work, in order to fulfil the assessment requirements.
- It is my responsibility to ensure that my submission is complete, correct and not corrupted.

Signature: Pakcheera Choppradit Date of signature: 8/10/21

(By submitting this form, you indicate that you agree with all the above text.)

COMP3009: Data Mining

Assignment

Pakcheera Choppradit 20303349

School of Elec Eng, Comp and Math Sci (EECMS),
Faculty of Science and Engineering
Curtin University

Due Date: Week 11 - Friday 8-October-2021, 12:00pm Perth time (mid day)

CONTENTS

Abstract

This report is for the assignment of data mining course COMP3009. The objective of assignment is to do data mining process including data preparation and analytic with binary classification problem. The dataset is not clean and require to solve the problems of irrelevant attributes, duplication, missing entries, wrong data type, data scaling and others. In this assignment, Python programming is used to approach all process. This assignment make me more understand about data mining process that the data preparation method is really important role and different models require different kind of preparation.

Introduction

The file name of dataset is "data2021.student". It is supervised learning with 1100 instances which contain 1000 labeled instances and 100 unlabeled instances and the attributes of classification is "Class" which is binary classification target. Sklearn and pandas is used to preparation data and building classification model and imblearn is used to solve the imbalance dataset. In this assignment aims to maximize the accuracy of prediction 100 instances which know that there are 50 number in each class. To achieve the goals, it require to adapt all knowledge that study in class and research more information on the internet to deeper understand whole process. Therefore, the explanation of all processes are written in this report.

Data preparation

Describing the data

At the beginning, I use **DataFrame.describe()** to see the overall of numeric attributed where ID attributed is set be the index. For Figure 1 below, it show that C4, C29 and C32 is contain null values and C15 and C17 have std equal to zero that mean it is all the same value.

	Class	C1	C4	C9	C15	C16	C17	C19	C20	C23	C25	C27	C29	C31	C32
count	1000.00000	1100.000000	1093.000000	1100.000000	1100.0	1100.000000	1100.0	1100.000000	1100.000000	1100.000000	1100.000000	1100.000000	1094.000000	1100.000000	5.000000
mean	0.27700	34.962727	20.347667	3265.750909	0.0	40530.608182	1.0	5001.148182	2.846364	1.401818	20.308182	2.978182	1.148995	3265.750909	3.000000
std	0.44774	11.345411	12.048965	2833.052110	0.0	28221.725221	0.0	1001.006037	1.104641	0.569466	12.037949	1.113846	0.356246	2833.052110	1.581139
min	0.00000	18.000000	3.000000	249.000000	0.0	1446.000000	1.0	2272.000000	1.000000	1.000000	3.000000	1.000000	1.000000	249.000000	1.000000
25%	0.00000	26.000000	11.000000	1366.000000	0.0	19447.750000	1.0	4326.500000	2.000000	1.000000	11.000000	2.000000	1.000000	1366.000000	2.000000
50%	0.00000	32.000000	18.000000	2301.500000	0.0	33598.000000	1.0	4969.000000	3.000000	1.000000	18.000000	3.000000	1.000000	2301.500000	3.000000
75%	1.00000	41.000000	24.000000	3967.250000	0.0	56142.000000	1.0	5677.000000	4.000000	2.000000	24.000000	4.000000	1.000000	3967.250000	4.000000
max	1.00000	75.000000	72.000000	18424.000000	0.0	220716.000000	1.0	8633.000000	4.000000	4.000000	72.000000	4.000000	2.000000	18424.000000	5.000000

Figure 1: Overall of numeric attributed

Next step is describe categorical part. In Figure 2 below, it show that C3,C11 and C13 is contain null values and C10 and C30 have unique equal to 1 that mean it is all the same value.

	Class	C2	C3	C5	C6	C7	C8	C10	C11	C12	C13	C14	C18	C21	C22	C24	C26	C28	C30
count	1000.0	1100	1093	1100	1100	1100	1100	1100	5	1100	1094	1100	1100	1100	1100	1100	1100	1100	1100
unique	2.0	2	5	10	5	4	4	1	3	4	3	4	3	5	2	3	5	4	1
top	0.0	yes	V3	V3	V1	V4	V4	F	V2	V4	V3	V3	V1	V2	V1	V2	V2	V3	T
freq	723.0	1058	370	304	670	437	366	1100	2	437	896	697	997	590	652	779	590	595	1100

Figure 2: Overall of category attributed

In conclusion, there are 1100 instance and 18 categorical, 14 numeric and 1 target attribute. Note that ID attributed is used to be index.

Irrelevant attributes

Removing relevant attributes help to improve running time of classification. For previous section we know that C10, C15, C17 and C30 have one unique values in each columns so I decided to drop all of this attributes. below is the function to delete irrelevant attributes.

```python
def drop_one_unique(_data):
  data=_data.copy()
  decs_cat=data.describe(exclude=[np.number]).T
  col_cat=decs_cat[decs_cat.unique==1].index
  data=data.drop(columns=col_cat)
  decs_num=data.describe().T
  col_num=decs_num[decs_num.std==0].index
  data=data.drop(columns=col_num)
  return data
```

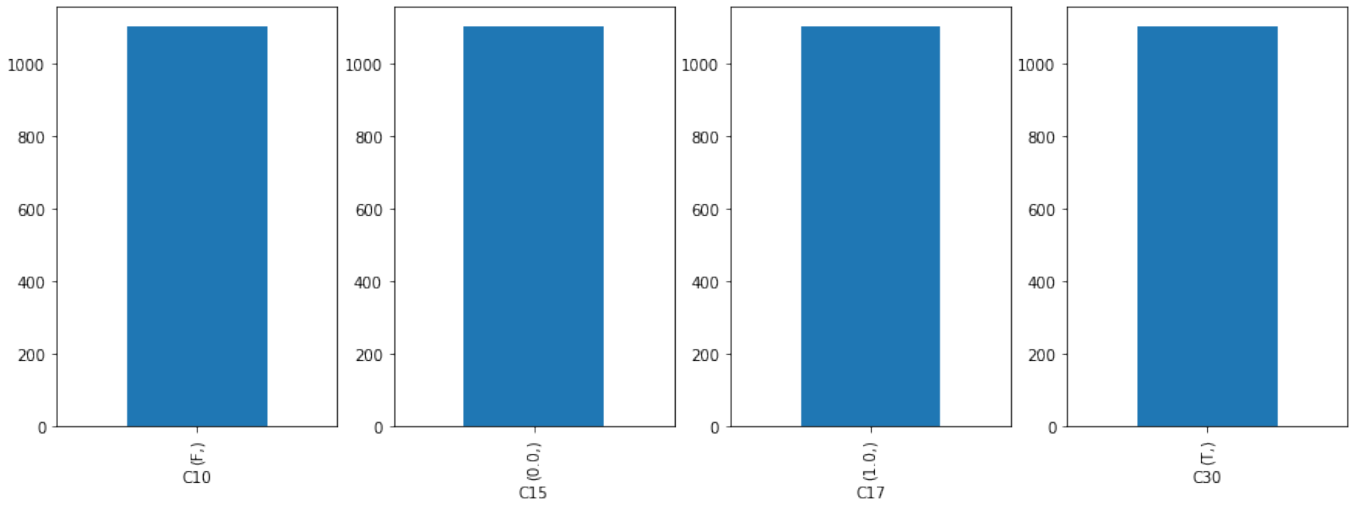

Figure 3: Bar plot of irrelevant attributed

Attributes	Data type	Reason for removing
C15 and C17	Numeric	Contain one value.
C10 and C30	Nominal	Contain one value.

Missing entries

It is general for real world dataset that will have missing values, the important role is how we deal with missing value. There are 3 methods to handle missing values as following:

1. **Impute with mean :** It is famous use for numeric data type.
2. **Impute with most frequent/most common class :** It is famous use for non-numeric data.
3. **Remove row or columns :** It use when there is a lot of missing values.

From describe the data section, we know that the data that contains null values is C3, C11, C13, C4, C29 and C32 (exclude Class). The action are describe for following rule:

1. If number of missing instances more than a half, the attributes will be dropped.

2. If number of missing instances less than a half, the attributed will be replace with most frequent value or mean values according to the type of data (mean when it is numeric and most frequent when it is non numeric)

The following rule will action under code below.

```
def drop_most_null(_data,precent=0.5):
  data=_data.copy()
  null_columns=data.columns[data.isna().any()].tolist()[1:]
  for col in null_columns:
    if data[col].isna().sum()>precent*len(data):
    data=data.drop(columns=col)
  return data
data=drop_most_null(data)
data["C3"]=data["C3"].fillna(data["C3"].mode()[0])
data["C4"]=data["C4"].fillna(data["C4"].mean())
data["C13"]=data["C13"].fillna(data["C13"].mode()[0])
data["C29"]=data["C29"].fillna(data["C29"].mode()[0])
```

Here is the table to summary action.

Attributes	Data type	Action
C3, C13 and C29	Nominal	Replace each with most frequent values (Rule2).
C4	Numeric	Replace with mean values (Rule2).
C11	Nominal	Remove columns (Rlue1).
C32	Numeric	Remove columns (Rlue1).

Duplication

In python **DataFrame.drop_duplicates()** can use for drop the duplicate row and we can apply to drop the duplicate column by transpose dataframe like this **DataFrame.T.drop_duplicates().T**.

Duplicate columns : C12 with C7, C31 with C9 and C26 with C21.
Duplicate row : row ID 901-1000 is duplicate with other

Data types

After I overlook the describe of dataset in picture below, the numeric data is in C1, C4, C9, C16, C19 and C25 and all of this is integer so I convert it from float to int64 and others attribute there are two choice to data convert data type which is object and category. I choose to convert the rest to category because it improve the speed and space with large set of data (Jeff, 2020).

Correlation

The multicollinearity can reduce the predict performance when the covarince between variable in training set and testing set is different for prevent this issue, I prefer to drop columns that have high correlation with other. In the picture below show that C25 is high correlation with C4. Therefore, I choose to drop C25.

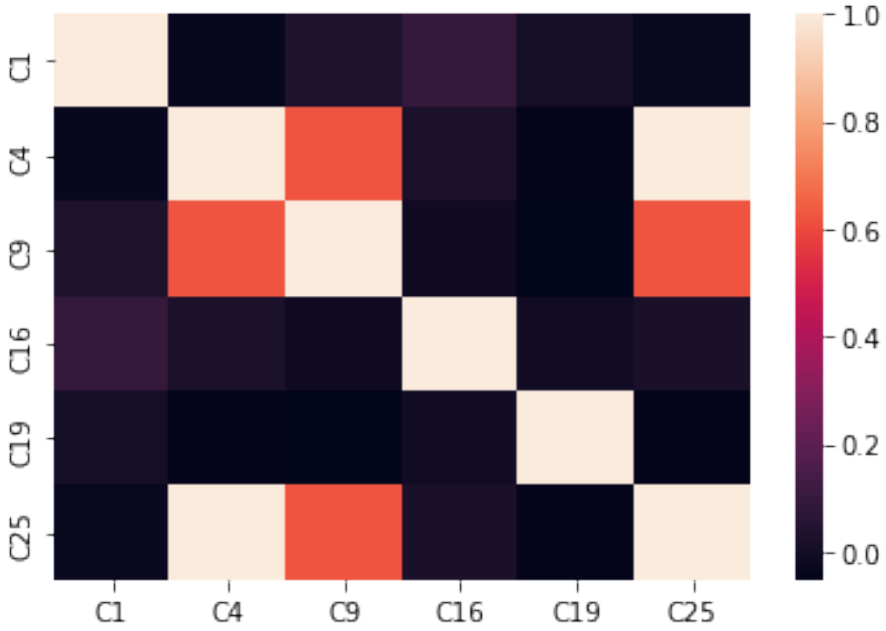

Figure 4: Correlation of numeric data

Delete outlier

When data have the outlier it will affect the performance of prediction. Therefore I decide to drop the row that have absolute Z-score > 3 of numeric type of data in training set it reduce form 900 instances to 851 instances.

Scaling

There are many method to scaling the data in this assignment. I choose MinMaxScalar() in numeric dataset because I plan to use MultinomialNB that not support the negative values. To make the data to normal distribution. I use log transformation before scaling. In MinMaxScalar(), use just train part to fit the function and transform in both train and test data part. Here is the formula of MinMaxScalar().

$$X' = \frac{x - min(x)}{max(x) - min(x)}$$

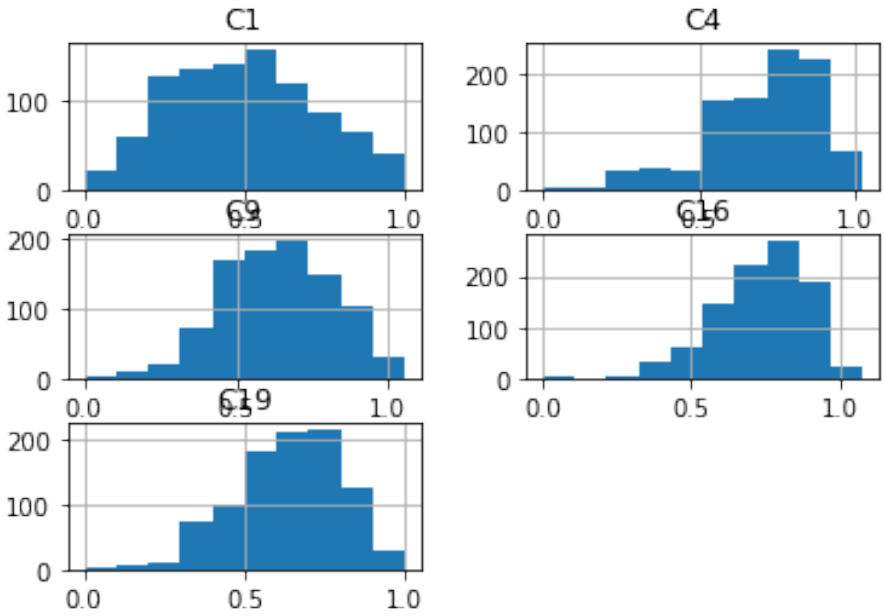

Figure 5: The histogram after scaling

LabelEncoding

KNN is the distance base classification if we convert nominal data to number like 0, 1 and 2. It will wrong meaning of distance between nominal data. To make distance between nominal data be the same. I decide to choose one hot encoding all nominal data that will be the int32 data type.

Label and unlabel split

In this process I do the train test split with label and unlabel data. Note that when I do any transform I fit in label data and transform to all data.

Data Classification

Imbalance issue

The imbalanced dataset is the huge affect the performance of machine learning. The good example is when there are 99% of negative data, if the model predict all to negative the accuracy will be 99% but it is not different to guess to the most class. There are many matrics to measure help this problem (will be mention later). Therefore, we should solve this problem.

There are 3 type of imbalanced learning, which is oversampling, undersampling and combine both. SMOTE is the famous method to oversampling data. I choose oversampling data because it will not

loss the information of data. The figure below is the bar char of number of Class before and after use SMOTE method.

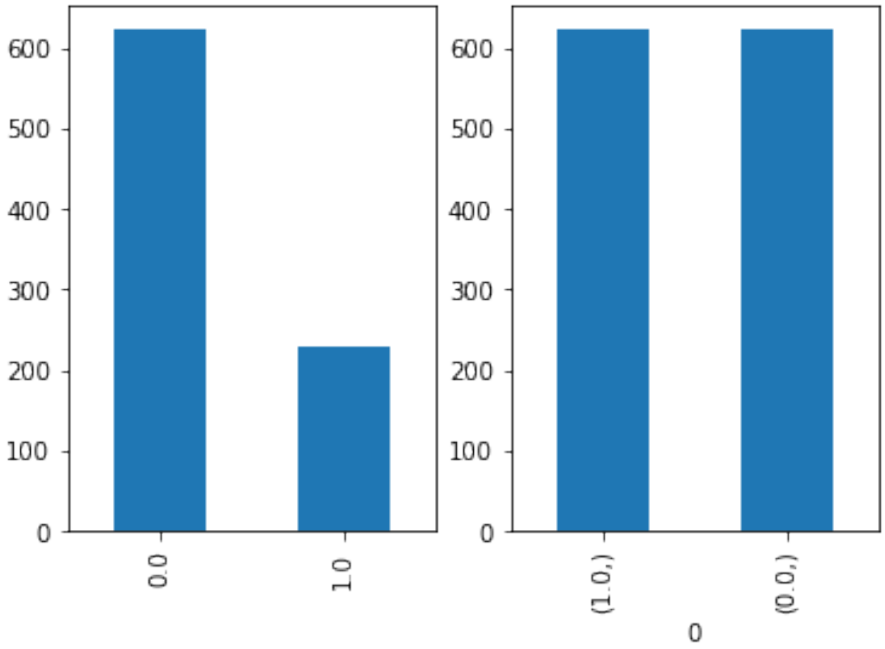

Figure 6: Before and after use SMOTE method.

Classifier Selection and Hyperparameter Tuning

In this process. I use imblearn.pipeline to create pipeline of SMOTE and GridSearchCV with StratifiedKFold to make sure that we will oversampling just train dataset and equal distribute the positive class in each fold. In GridSearchCV I tuning the balanced_accuracy metrics because there are not equal positive and negative class in test set (balance just train data to prevent overfitting) to more information about metric will be talked in other section.

Classifier Selection and its parameter

1. K Nearest Neighbors (KNN)
It is the simplest supervised machine learning algorithm. The concept of this algorithm is distance. We can choose number of n_neighbors which is the number of neighbors and the distance method defined as p. When p=1,2 and 3 is Manhattan distance, Euckidean distance and Minkowski distance, respectively.

2. Decision Tree
It use method of recursive partitioning to break the data in to part. There are two function to split tree is gini and entropy and the max_depth which is the max of depth in model parameter is the importance parameter to prevent the overfitting.

3. Random Forest
It is the extension of decision tree. The algorithm is build many of decision trees train on partial train and bagging all tree and decide the last answer with voting. The parameter is the mostly same as decision tree. There are other parameter like number of tree but in this assignment I use the default values.

4. Multinomial Naive Bayes
This is use Bayes's theorem that base on multinomial distribution there is one parameter that is alpha meaning smooth parameter (0 for parameter).

Hyperparameter Tuning

The metric score that I use is balanced_accuracy which the formula equal $\frac{TPR+TNR}{2}$ because the validation set is imbalanced but we know that th test data set have equal positive and negative number. The following table is the result of hyperparameter with GridSearchCV.

Model	parameter	best balanced_accuracy	best parameter
K Nearest Neighbors	n_neighbors, p	67.54	n_neighbors=27, p=1
Decision Tree	criteriona, max_depth	67.68	criteriona= gini, max_depth=4
Random Forest	criteriona, max_depth	69.32	criteriona= entropy, max_depth=6
MultinomialNB	alpha	70.13	alpha=1e-08.

K Nearest Neighbors : p = 1 mean use Manhattan distance and n_neighbors=27
Decision tree : use gini with max_depth=4
Random Forest : use entropy with max_depth=4
MultinomialNB : use alpha= 1e-08

Classifier comparison

I do the StratifiedKFold cv=10 with all model under the same oversampling in repetition with 20 times and plot the box plot to comparison. The metric that I use is AUC_ROC curve, balanced_accuracy and F1. AUC_ROC curve is the plot of TPR and FPR where TPR id y-axis and FPR is x-axis. It will be good measure seperability when the values is close to 1 (Narkhede, 2021). F1 is focus on positve class, the equaltion is equal to

$$\frac{2 * precision * recall}{precision + recall}$$

I use F1 score because I want to know how well we predict the positive class with SMOTE method. The balanced_accuracy is equal weighted both class equally. This is the good metric to use for estimate the accuracy of unseen dataset because we know that each class have same number of instances.

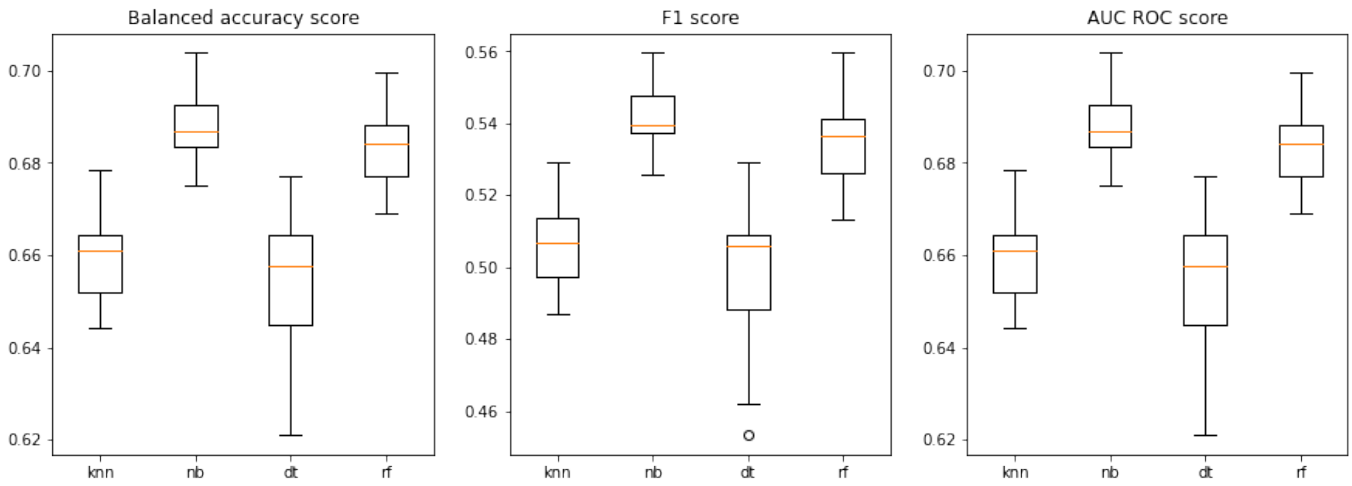

Figure 7: Box plot show score of 20 times repetition.

The above picture is show the 20 times of StratifiedKFold cv=10 of F1, AUC ROC and balanced accuracy. It is obviously that MultinomialNB and Random forest is the first two most score in AUC ROC ,balanced accuracy and F1. Let see the mean of 20 times in table below.

Model	AUC ROC	Balanced accuracy	F1
K Nearest Neighbors	65.94%	65.94%	50.66%
Decision Tree	65.38%	65.38%	49.80%
Random Forest	68.30%	68.30%	53.50%
MultinomialNB	68.81%	68.81%	54.19%

Prediction

In final model I choose MultinomialNB and Random Forest, the process is in following process:

1. Clean data both labels and unlabels data (fit in label set and transform in all data).
2. Do oversampling SMOTE method om labels data set.
3. Fit the both models on labels data. (X is whole columns except Class and y is Class columns).
4. Predict Class on unlabels set (drop Class of unlabel set out).
5. Save predict values.

The estimate accuracy of MultinomialNB and Random Forest are 68.

Random Forest ratio of predict is 43:57 (positvie(1) : negative(0))

MultinomialNB ratio of predict is 56:44 (positvie(1) : negative(0))

References

Jeff. (2020, December 11). "Pandas difference between object and category" Code answer. Dizzy Coding. Retrieved October 7, 2021, from http://dizzycoding.com/pandas-difference-between-object-and-category-code-answer/.

Narkhede, S. (2021, June 15). Understanding AUC - roc curve. Medium. Retrieved October 7, 2021, from https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5.