SIT307 Machine Learning
Mushroom Binary Classification Tasks

CHANPUTHI TITH
219498222
ctith@deakin.edu.au

1.) Background

Mushroom is an important ingredient for most species to consume including human. However, there are some types of mushrooms which we can consume and some we cannot.

In this study we will analyse different mushroom's family which is editable or poisonous with a few binary classification algorithms based on research paper "Mushroom data creation, curation, and simulation to support classification tasks" written by Dennis Wagner, Dominik Heider and George Hattab.

This report will begin by exploring the primary (or 1987) dataset and the secondary (or 2020) dataset both provided on the University of California, Irvine (UCI) Machine Learning Repository. After that, we will illustrate the pre-processing steps including dealing with missing values and data encoding. Then, we will split both datasets into training and testing dataset respectively. Finally, we will apply four machine learnings algorithm on both datasets including Navies Bayes, Logistic Regression, Linear Discriminant Analysis, and Random Forest Classifier. We will use Accuracy score and F2 score to report the performance by visualizing it as Box Plot.

2.) Explore Dataset

There are two datasets in this study including Primary dataset and Secondary dataset namely, primary_data.csv and secondary_data.csv. The information of both dataset is demonstrated below on each section.

Primary Dataset

The Primary Dataset consist of 173 rows (or instances) and 23 columns including 22 features and 1 target variable. The information of this dataset is shown below.

However, there are some features which are quantitative variables including cap-diameter, stem-height, and stem-width. These features consist of minimum and maximum numerical value. Thus, we have converted it into float data type by combined the minimum value and maximum value together and divide it by two to receive its mean (or average) value.

Column Number	Column Name	Data Type
0	family	object
1	name	object
2	class	object
3	cap-diameter	object -> float
4	cap-shape	object
5	cap-surface	object
6	cap-color	object
7	does-bruise-or-bleed	object
8	gill-attachment	object
9	gill-spacing	object
10	gill-color	object
11	stem-height	object -> float
12	stem-width	object -> float
13	stem-root	object
14	stem-surface	object
15	stem-color	object
16	veil-type	object
17	veil-color	object
18	has-ring	object
19	ring-type	object
20	spore-print-color	object
21	habitat	object
22	season	object

Beside explore dataset information, we also found the quantitative number and percentage of each class, and the auto correlation between each three of numerical variable as well.

This primary dataset consists of 77 editable mushrooms, and 96 poisonous mushrooms which equivalent to 44.5% and 55.5% respectively.

	cap-diameter	stem-height	stem-width
cap-diameter	1.00	0.42	0.71
stem-height	0.42	1.00	0.43
stem-width	0.71	0.43	1.00

Secondary Dataset

The Secondary Dataset consist of 61069 rows (or instances) and 21 columns including 20 features and 1 target variable. Unlike Primary dataset, this dataset has no datatypes issue for any variable. The information of this dataset is shown below.

Column Number	Column Name	Data Type
0	class	object
1	cap-diameter	object
2	cap-shape	object
3	cap-surface	object
4	cap-color	object
5	does-bruise-or-bleed	object
6	gill-attachment	object
7	gill-spacing	object
8	gill-color	object
9	stem-height	object
10	stem-width	object
11	stem-root	object
12	stem-surface	object
13	stem-color	object
14	veil-type	object
15	veil-color	object
16	has-ring	object
17	ring-type	object
18	spore-print-color	object
19	habitat	object
20	season	object

Beside explore dataset information, we also found the quantitative number and percentage of each class, and the auto correlation between each three of numerical variable as well.

This secondary dataset consists of 27181 editable mushrooms, and 33888 poisonous mushrooms which equivalent to 44.5% and 55.5% respectively.

	cap-diameter	stem-height	stem-width
cap-diameter	1.00	0.42	0.69
stem-height	0.42	1.00	0.43
stem-width	0.69	0.43	1.00

3.) Dealing with Missing Values

To deal with missing values, we have performed two steps. First, we will remove any variables which have missing value larger than 50%. Second, we will replace the remaining missing values with its most frequent value using mode.

Step 1: Remove variable with larger missing values

We have founded that there are five variables which have larger (more than 50%) missing values for both Primary dataset and Secondary dataset. These variables are stem-root, stem-surface, veil-type, veil-color, and spore-print-color.

Step 2: Replace remaining missing value with mode

After removed five larger missing value variables, there are four variables which have little (less than 50%) missing value in both datasets. Those variables are cap-surface, gill-attachment, gill-spacing, and ring-type. We replaced the missing values of these four variables with its most frequent values using mode.

4.) Data Transformation and Encoding

To make prediction or classification with machine learning algorithms/models, data must be numerical value.

In this data transformation and encoding section, we convert each variable into numerical value based on its datatype as shown in Pseudo code below.

```
FOR each column in the dataset:
    IF datatype is integer or float:
        continue (or skip it)
    ELIF datatype is binary value:
        Encode binary variable using LabelEncoder
    ELSE:
        Encode the remaining nominal variable using OneHotEncoder
```

Based on our finding, there are three quantitative variables such as cap-diameter, stem-height, and stem-width, and three binary variables including class, does-bruise-or-bleed, and has-ring. These variables are unchanged or encoded into numerical value using label encoder respectively while the remaining nominal variable are encoded using one hot encoder.

After completed pre-processing data variables, both datasets could be visualized with heatmap to understand the correlation between each variable as below.

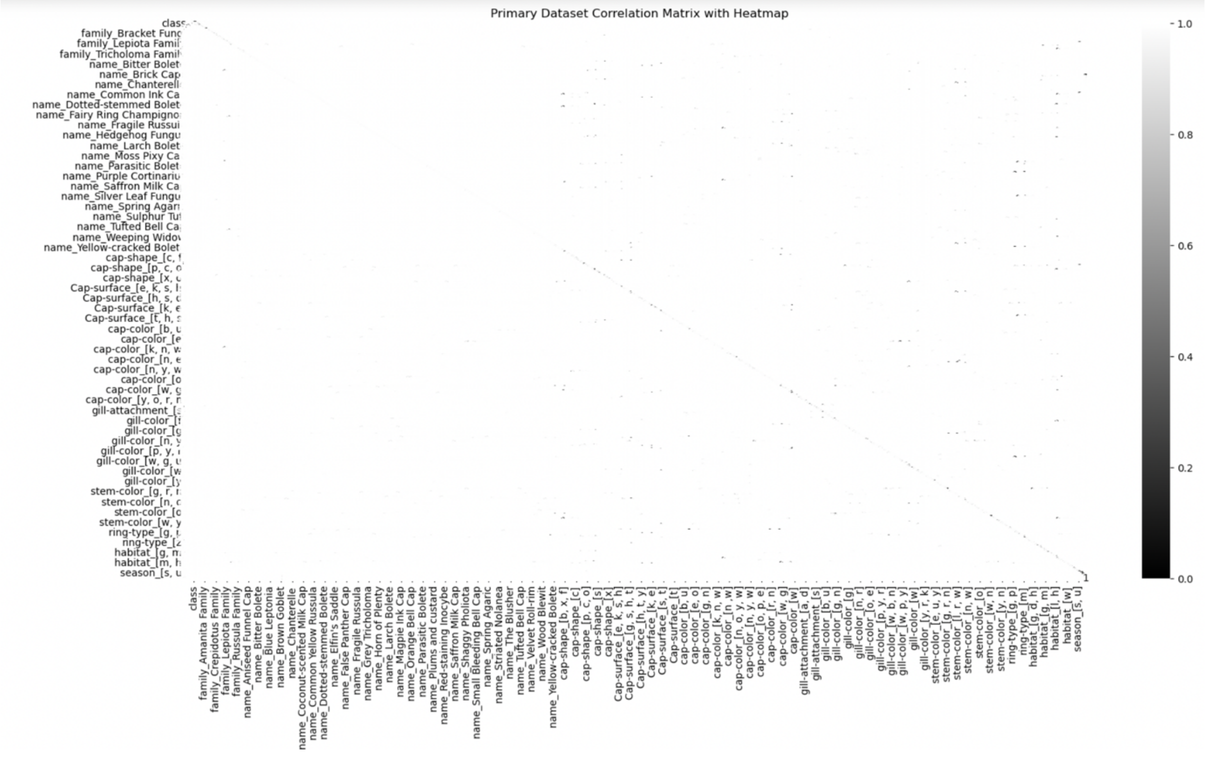

Figure 1: Primary Dataset Correlation Matrix with Heatmap

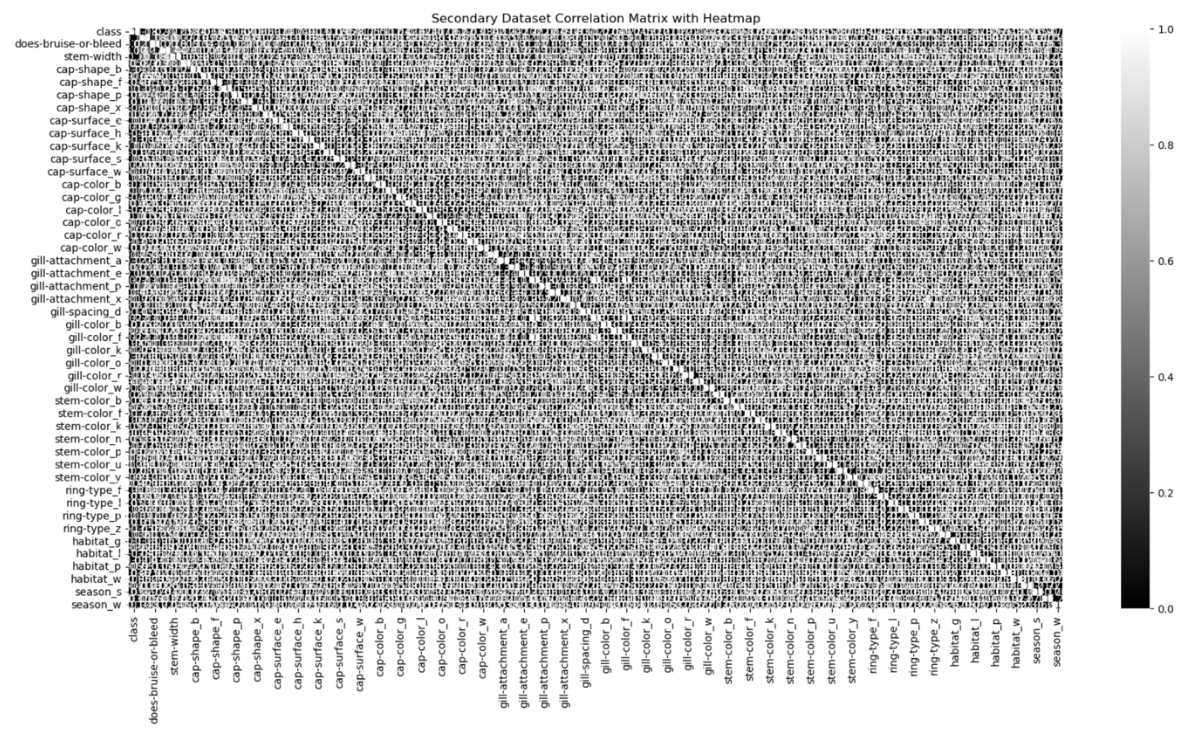

Figure 2: Secondary Dataset Correlation Matrix with Heat Map

5.) Splitting Dataset

Both datasets are randomly split into training and testing data set with a ratio of 80% and 20% of the data respectively based on the standard Pareto principle.

After splitting the data, the primary dataset has a training set of 138 variables and a testing set of 35 variables. Whereas the secondary dataset has a training set of 48855 variables and a testing set of 12214 variables respectively.

6.) Binary Classification Models and Performances

In this study, we applied four machine learning classification algorithms including Naïve Bayes, Logistic Regression, Linear Discriminant Analysis, and Random Forest Classifier.

```
gnb_model = GaussianNB()
logistic_model = LogisticRegression()
lda_model = LinearDiscriminantAnalysis()
rf_model = RandomForestClassifier()
```

To make it simple and efficient, I have created a function namely, classification_performance, to report performances of each machine learning models with accuracy score and F2 score metrics by taking X_train, y_train, and model as input. This function is shown below.

```
# classification performance with cross validation
def classification_performance(X_train, y_train, model):
    # Define K-fold for cross validation, and f2 scoring
    accuracy_scores = []
    f2_scores = []
    kf = KFold(n_splits=5, shuffle=True, random_state=42)
    f2 = make_scorer(fbeta_score, beta=2)
    # Perform cross validation for prediction
    accuracy_scores = cross_val_score(model, X_train, y_train, cv=kf, scoring="accuracy")
    f2_scores = cross_val_score(model, X_train, y_train, cv=kf, scoring=f2)
    # Add Accuracy and F2 scores for BoxPlot
    acc_scoresList.append(accuracy_scores)
    f2_scoresList.append(f2_scores)
    # Accuracy and F2 Result for each ML models
    print('Accuracy Score:', accuracy_scores.mean())
    print('F2 Score:', f2_scores.mean())
```

After applied classification performance function on each machine learning models, the performance score on both dataset is shown in the table and box plot (figure 3) below.

	Primary Dataset		Secondary Dataset	
	Accuracy	F2	Accuracy	F2
Naïve Bayes	0.52	0.46	0.72	0.77
Logistic	0.56	0.65	0.78	0.79
LDA	0.55	0.70	0.78	0.79
Random Forest	0.57	0.72	1.00	1.00

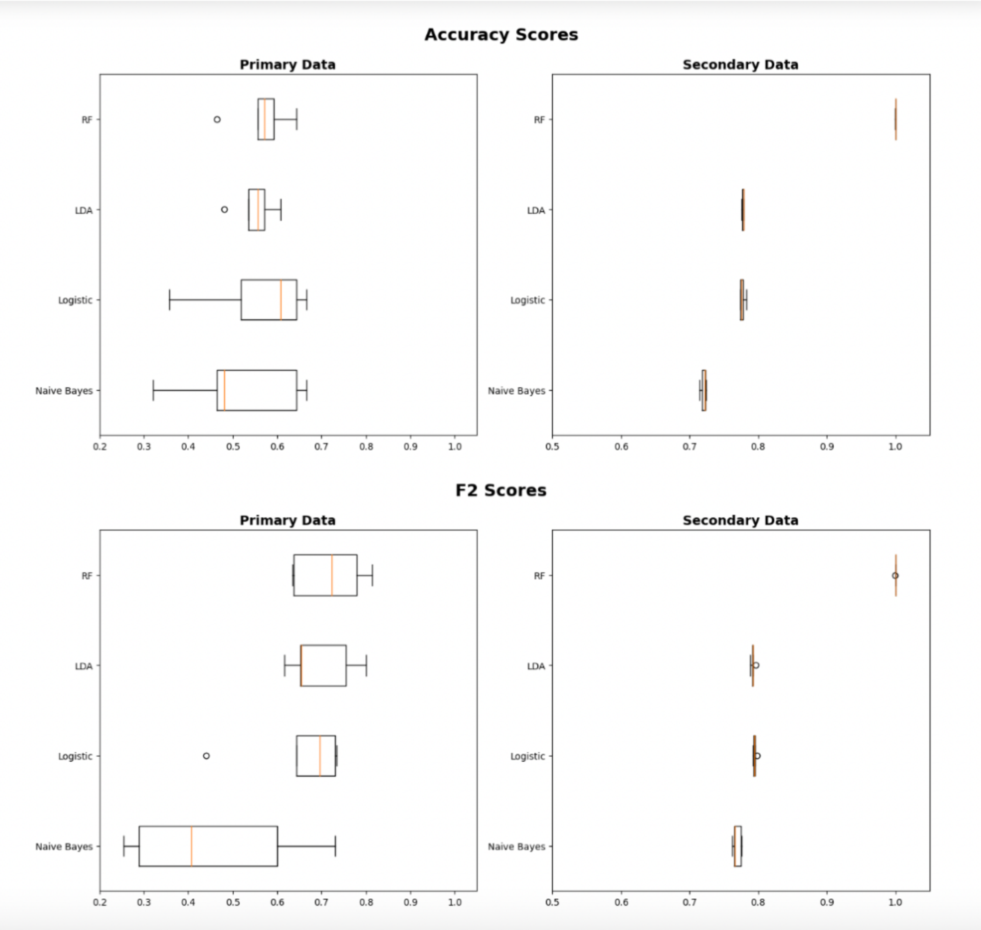

Figure 3: Accuracy and F2 Scores of ML models on both dataset

7.) Conclusion

In conclusion, this study has found that random forest classifier is the best model among the four implemented model with an accuracy score of 0.57 and f2 score of 0.72 on primary dataset, and a near perfect score of 0.999 (almost 1.00) on secondary dataset.

By comparing both dataset, secondary dataset has better accuracy score and f2 score for all of the machine learning algorithms with scores of more than 0.70. In contrast, primary dataset has poor performance score less than 0.75 for every machine learning algorithm. This could be due to the complexity and quality of the dataset because the primary dataset is made in 1987.