ML Project 1
Mushroom Classification using Naive Bayes Algorithm

Akshay K (23BM6JP05)
N Surya Prakash Reddy (20CS10038)

1 Introduction

The objective of this project is to classify mushrooms into 'poisonous' or 'edible' using the Naive Bayes classifier Algorithm.

Sklearn's CategoricalNB tool has been used for comparison and Laplace smoothing parameter(α) has been used as a hyper-parameter.

2 Implementation

2.1 Train Test Split

The dataset is split into 80:20 train-test-split by using $train_test_split$ method from sklearn library.

2.2 Fit

The training data set is used to calculate conditional probabilities of the Features(X) and the prior probabilities of Labels(Y).

When the hyper-parameter α is given:

- $P(Y) = \frac{count(y)+\alpha}{N+\alpha.|C|}$, where N is number of rows and C is number of labels.
- $P(x_i|Y) = \frac{count(x_i,y)+\alpha}{count(y)+\alpha.|V|}$, where V is the number of unique values of x_i.

The smoothing parameter is used to prevent the probability of unseen features in the training dataset being zero. This avoids the overall probability being zero while testing.

2.3 Predict

The testing data is then passed to the predict method which makes use of the proportionality

- $P(Y|X)\ \alpha\ \prod P(x_i|Y)\ P(Y)$

The label with the highest prediction value will be output as the class for the given set of features in X.

2.4 Reporting

The classification_report method from sklearn has been used for reporting precision, recall and f1-score and accuracy_score method has been used to report accuracy.

3 Results

3.1 Observations

We have implemented an auxiliary method that keeps track of the Features and Labels in the training set for which probabilities are zero when alpha is zero. Using these we searched the test split for occurrences of such data and found that there were no existing rows with those values.

This means that the whole dataset does not contain the unseen data and would cause the effect where smoothing reduces the accuracy since the dataset favours over-fitting.

3.2 Custom Implementation

The reported values for test data set are

1. $\alpha = 0$, Accuracy = 1.00

	precision	recall	f1-score
e	1.00	1.00	1.00
p	1.00	1.00	1.00

2. $\alpha = 0.1$, Accuracy = 0.985

	precision	recall	f1-score
e	0.97	1.00	0.99
p	1.00	0.97	0.98

3. $\alpha = 0.5$, Accuracy = 0.962

	precision	recall	f1-score
e	0.94	1.00	0.97
p	1.00	0.92	0.96

4. $\alpha = 1$, Accuracy = 0.953

	precision	recall	f1-score
e	0.92	1.00	0.96
p	1.00	0.91	0.95

3.3 Sklearn Classifier

The reported values for test data set are

1. $\alpha = 0$, Accuracy = 1.00

	precision	recall	f1-score
e	1.00	1.00	1.00
p	1.00	1.00	1.00

2. $\alpha = 0.1$, Accuracy = 0.985

	precision	recall	f1-score
e	0.97	1.00	0.99
p	1.00	0.97	0.98

3. $\alpha = 0.5$, Accuracy = 0.962

	precision	recall	f1-score
e	0.94	1.00	0.97
p	1.00	0.92	0.96

4. $\alpha = 1$, Accuracy = 0.953

	precision	recall	f1-score
e	0.92	1.00	0.96
p	1.00	0.91	0.95

We can see that both the models have similar values of precision, recall, f1-score and accuracy which decrease as α increases.