Breast Cancer Detection using Naïve Bayes

Shavin K, Varun M, Venkat Satish A, Venkatesh S
1MS18CS110, 1MS18CS133, 1MS18CS134, 1MS18CS135

- **Description of Dataset:**

This breast cancer dataset titled "Wisconsin Breast Cancer Database (January 8, 1991)" was obtained from UCI Repository uploaded by the University of Wisconsin Hospitals, Madison from Dr. William H. Wolberg. There are 699 instances in this dataset, each consisting of 10 attribute values plus the class attribute as follows:

Attribute	Domain
Sample Code Number	ID Number
Clump Thickness	1 – 10
Uniformity of Cell Size	1 – 10
Uniformity of Cell Shape	1 – 10
Marginal Adhesion	1 – 10
Single Epithelial Cell Size	1 – 10
Bare Nuclei	1 – 10
Bland Chromatin	1 – 10
Normal Nucleoli	1 – 10
Mitoses	1 – 10
Class	2 for Benign (65.5%) 4 for Malignant (34.5%)

Link to the dataset: https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Diagnostic)

- **Data Pre-processing:**

1) Missing Values:

There were 16 missing values in this dataset which were represented as '?', all of which belonged to 'Bare Nuclei' column. We replaced all the missing values (represented as yellow stripes in Fig. 1.1) with the median of the particular column for the corresponding class. We chose median as its resistant to outliers and results in integer values which is compatible with the other values. As seen in Fig. 1.2, after the operation, no more missing values are present.

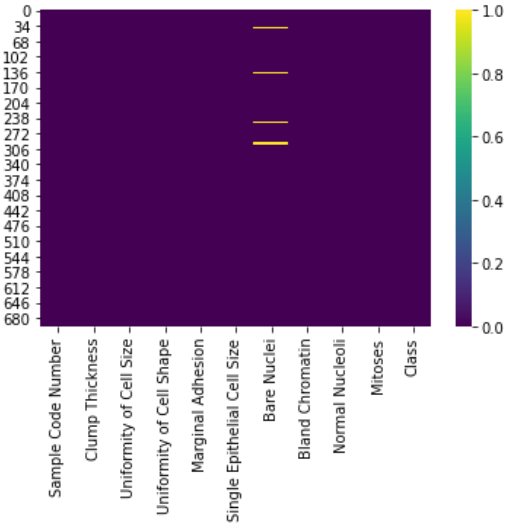

Fig. 1.1. With missing values

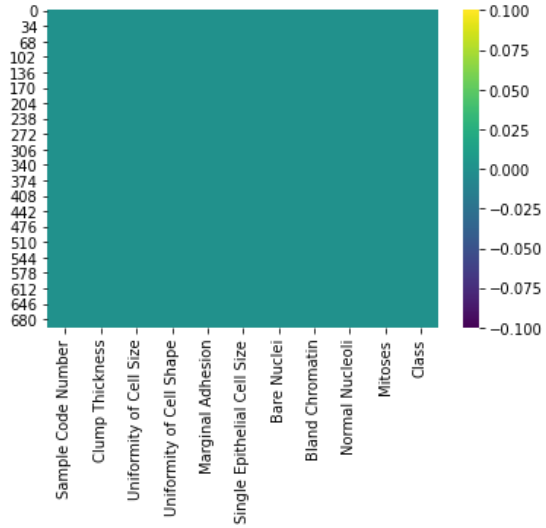

Fig. 1.2. After replacing missing values

2) Normalization:

We used the in-built StandardScaler module for normalization which is based on Z-score transformation. It transforms each data value within a range of -3 to +3. We perform this operation as columns with differing range of values can make it harder for the model to learn as well as to make the below data pre-processing steps more effective.

3) Handle Outliers:

Found the outliers using the Inter-Quartile Range (IQR) method and replaced it with the median of the corresponding column. We chose median over removing the outliers because many of the outliers belonged to Malignant class which already had fewer tuples than Benign class. Removing them would make it harder for the model to learn the Malignant class category well.

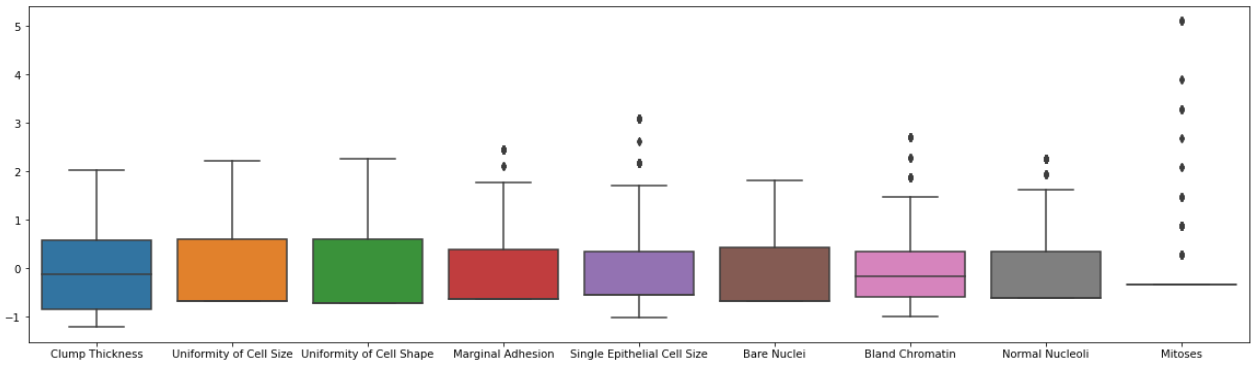

Fig. 2.1. Box Plot for the dataset with outliers

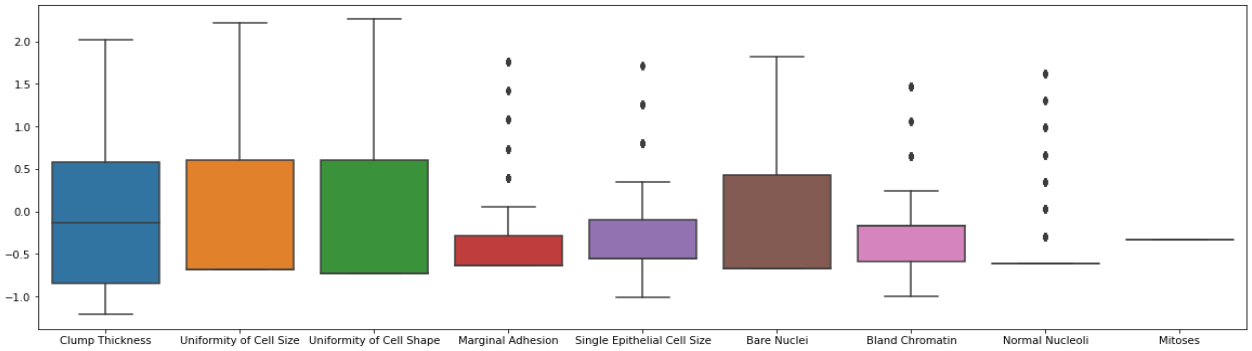

Fig. 2.2. Box Plot after replacing outliers

We used the Box Plot module from Seaborn library to visualize the outliers as shown above. In Fig. 2.1, there are many columns where values go up to 5 (i.e., outliers according to IQR method). After replacing them with the median, the values stay well under 2 as indicated in Fig. 2.2.

4) Handle Skewness:

Skewness is the measure of how much the probability distribution of a random variable deviates from the normal distribution. We found that except Mitoses, all other columns had right skewness, some of which is indicated in Fig. 3.1(a), Fig. 3.2(a) and Fig. 3.3(a). Hence, we applied the appropriate transformations such as cube root, inverse cube root and reciprocal to each column in order to reduce skewness. Some examples for the effect of the transformations are shown in Fig. 3.1(b), Fig. 3.2(b) and Fig. 3.3(b) where the columns' distributions have been transformed to look more like a Bell-shaped curve. This makes it easier for the model to fit the dataset better.

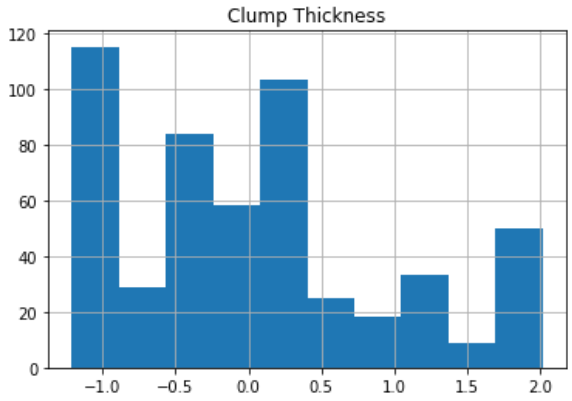

Fig. 3.1(a). With skewness

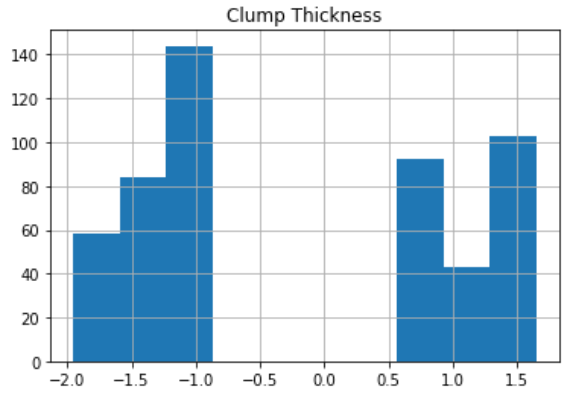

Fig. 3.1(b). After eliminating skewness

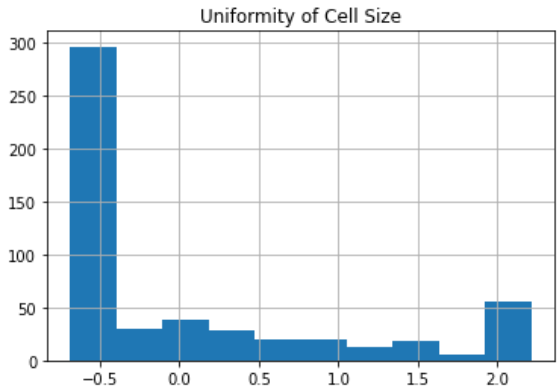

Fig. 3.2(a). With skewness

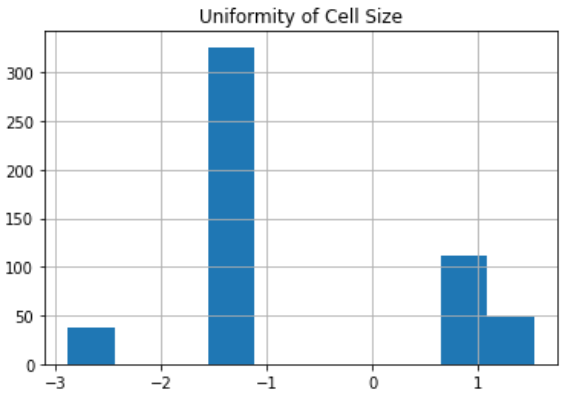

Fig. 3.2(b). After eliminating skewness

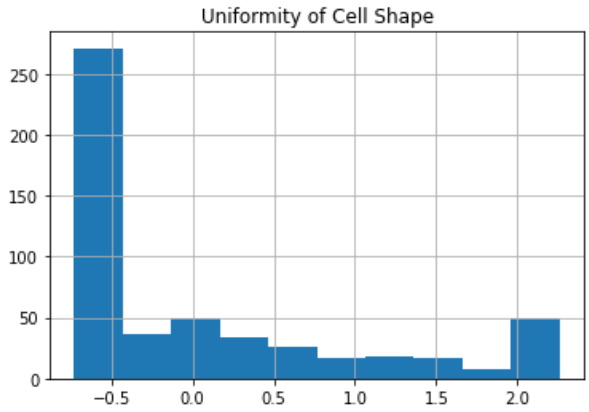

Fig. 3.3(a). With skewness

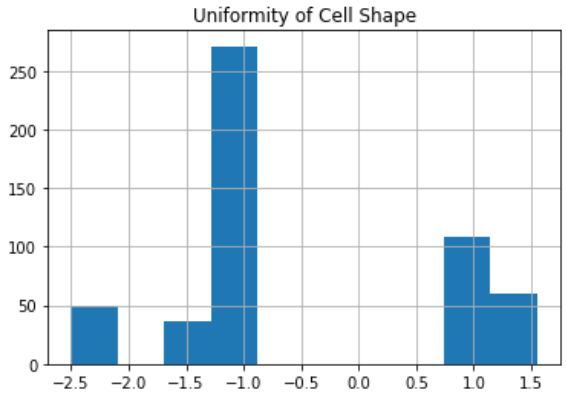

Fig. 3.3(b). After eliminating skewness

5) Feature Selection:

As all the features in a dataset may not be equally important in decision making, we perform feature selection to select the best features. Tree-based classification algorithms such as ExtraTreesClassifier inherently keep track of the best features for a given dataset. Thus, by fitting such a classifier to our dataset, we obtain the best features which has been plotted and shown in Fig. 4.1. In Fig. 4.2, we compute the correlation matrix which shows the correlation between a given column and every other column in a dataset. We noticed that 'Mitoses' and 'Bland Chromatin' always ranked low in feature importance in both the below methods. Hence, we removed them from the dataset.

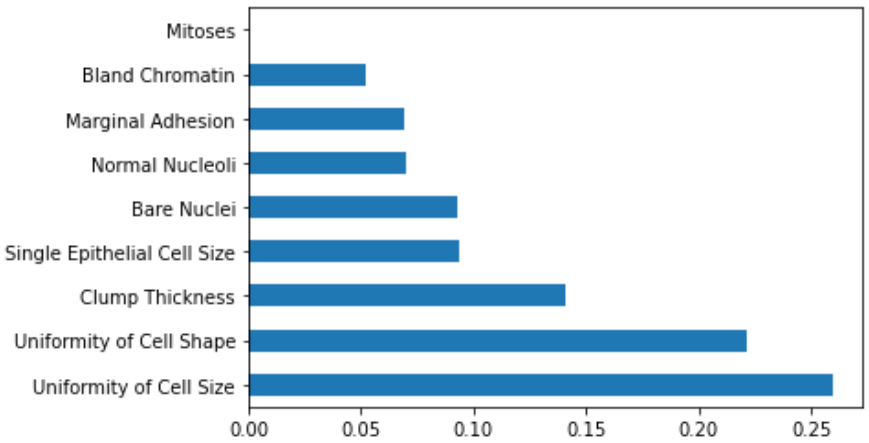

Fig. 4.1. Feature Selection by ExtraTreesClassifier

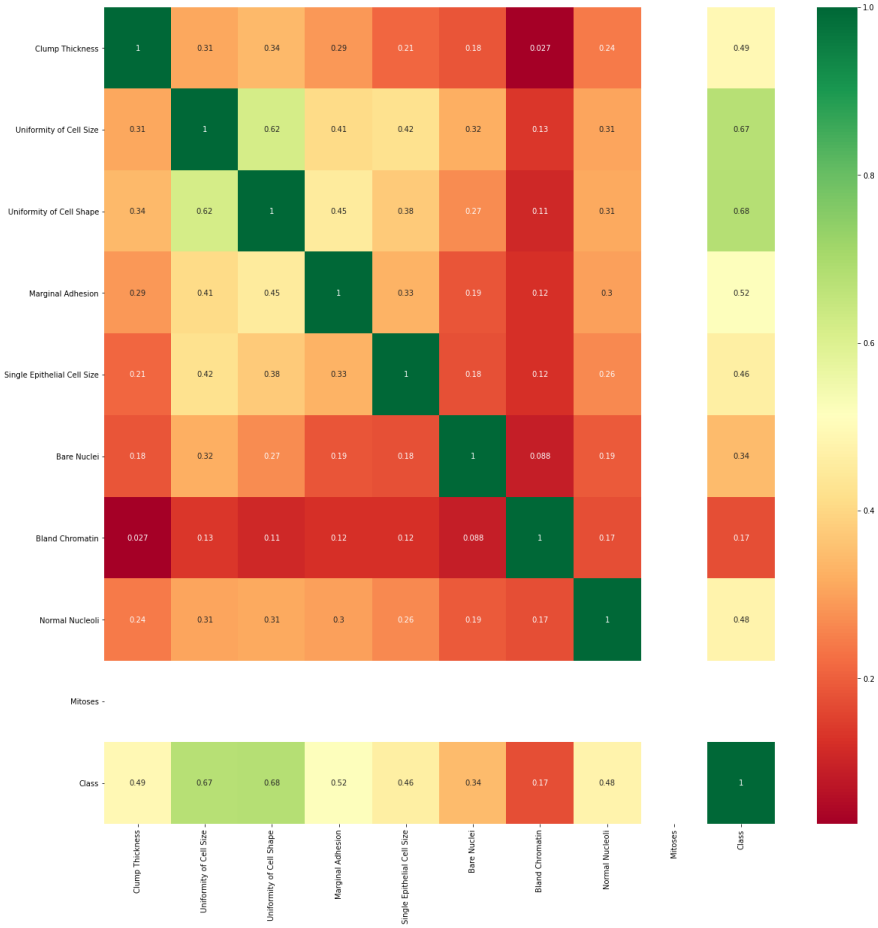

Fig. 4.2. Feature Selection by Correlation Matrix

6) <u>Dimensionality Reduction using Principal Component Analysis (PCA):</u>

PCA is a dimensionality reduction technique which identifies pattern in data by detecting the correlation between the variables. It finds the direction of maximum variance in higher dimension data and transforms it into a smaller dimension subspace. The purpose of the above operation was to reduce overfitting and to visualize the results in a graphical manner.

- **<u>Methodology and Algorithm:</u>**

Naive Bayes classifier is a probabilistic machine learning model that's used for classification tasks. It is not a single algorithm but a family of algorithms where all of them share a common principle which is that every pair of features being classified is independent of each other, i.e., presence of one particular feature does not affect the other. Hence the name, Naive. The core of the classifier is based on the Bayes Rule:

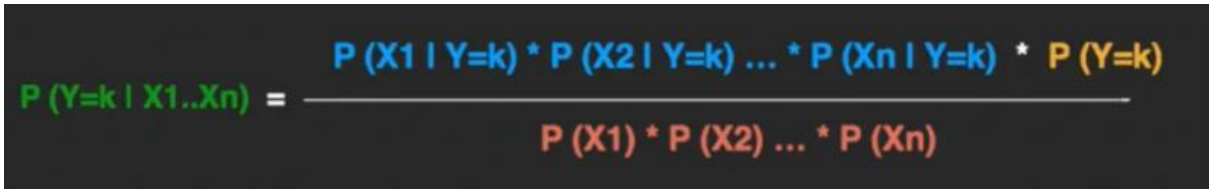

Fig. 5. Bayes Rule

The Naïve Bayes classifier works by computing the probability that an input example belongs to a class 'k' (for all possible 'k' values), given the various features of that example $X_1, X_2, \ldots\ldots, X_n$. As indicated in Fig. 5, due to the assumption of independence of features, we can simply take the product of probabilities of each feature (as shown in the denominator) as well as the product of probabilities of each feature given that it belongs to the class 'k' (for all 'k' values, as shown in the numerator). At the end, we take the maximum probability value across all classes 'k' to determine the final classification by the classifier.

- ## Results:

1) Before Hyperparameter Tuning:

Accuracy: 95.42 %

	Precision	**Recall**	**F-Score**
Benign	0.990566	0.937500	0.963303
Malignant	0.898551	0.984127	0.939394

The accuracy and F-score obtained above is pretty good for cancer detection using Naïve Bayes classifier after performing all the discussed data pre-processing. However, our model does perform better for Benign class than Malignant class as the dataset was skewed towards Benign class. Getting more tuples for Malignant class will help in improving its F-score value.

2) After Hyperparameter Tuning:

Accuracy: 96.54 %

	Precision	**Recall**	**F-Score**
Benign	0.981818	0.964286	0.972973
Malignant	0.938462	0.968254	0.953125

We performed hyperparameter tuning using GridSearchCV for the 'var_smoothing' hyperparameter to obtain its optimal value. 'var_smoothing' refers to the portion of the largest variance of all features that is added to variances for calculation stability. We found the optimal value to be 1.5. Doing so resulted in an improved accuracy and F-score for our model.

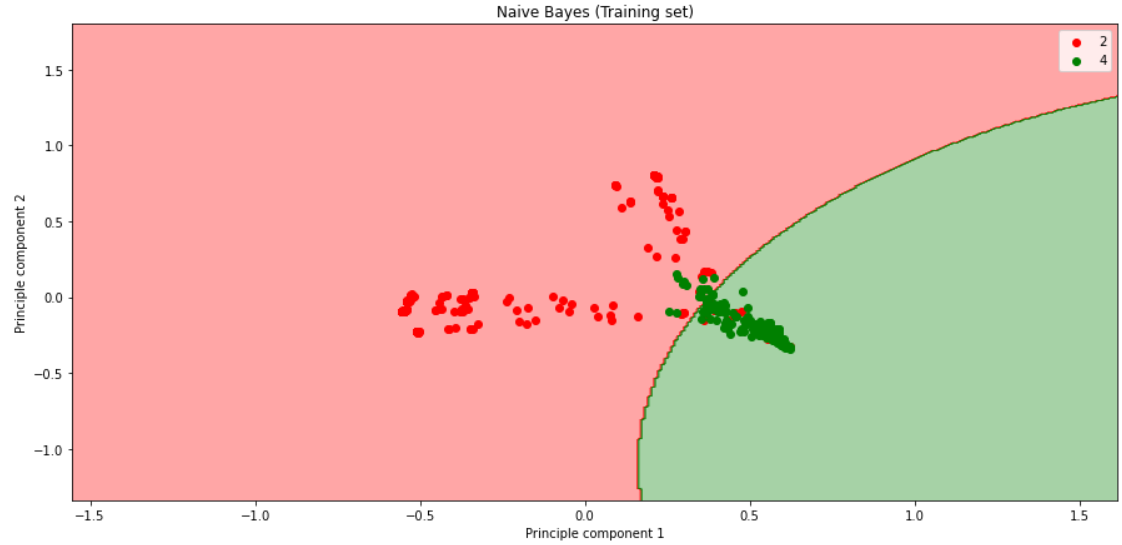

Fig. 6.1. Visualising training set results (‘2’ indicates Benign class and ‘4’ indicates Malignant class)

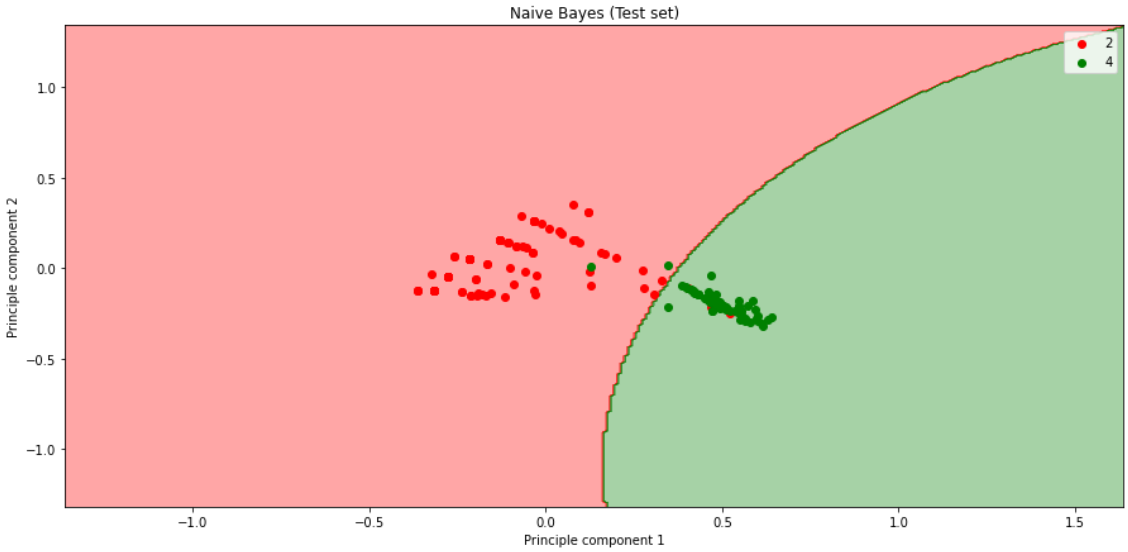

Fig. 6.2. Visualising test set results (‘2’ indicates Benign class and ‘4’ indicates Malignant class)

In both Fig. 6.1 and Fig. 6.2, the red region indicates Benign class and the green region indicates Malignant class. The red dots indicate that the model predicted those tuples as Benign and the green dots indicate the model predicted them as Malignant. The black, curved line is the decision boundary generated which separates the 2 classes. Hence, we can see that for this dataset, the Naïve Bayes classifier is a non-linear classifier.