Algorithm 3. Bayesian Linear Regression

Bayesian regression is a statistical approach to estimating the parameters of a linear model by incorporating Bayesian principles. Unlike traditional linear regression, it doesn't assume a specific data distribution. Bayesian regression places prior probability distributions on parameters, enabling the expression of prior beliefs about the data, leading to complex models. It excels in accurate predictions with limited data and provides uncertainty estimates. In contrast, traditional linear regression is simpler and faster but relies on the assumption of a specific data distribution. The choice between them depends on the model's complexity and the validity of data assumptions, with Bayesian regression offering greater flexibility and richer information but being more intricate to implement.

Implementation of Bayesian Linear Regression

In Bayesian linear regression, we have a dataset with independent variable X and dependent variable Y. We want to find the posterior distribution of the regression coefficients. The model can be represented as:

$$Y = Xb + \varepsilon$$

Where

1) Y is the matrix of the observed target values.
2) X is the matrix where each row represents an observation and it contains features.
3) b is a matrix containing the regression coefficients.
4) ε is the error term with a normal distribution $\varepsilon \sim N(0, 1/\beta)$, where β is the precision of the error term.

Posterior Distribution:

To find the posterior distribution of the regression coefficients, we use Bayes' theorem:
$P(b \mid X, Y, \alpha, \beta) \propto P(Y \mid X, b, \beta) * P(b \mid \alpha)$

1) $P(b \mid X, Y, \alpha, \beta)$ is the posterior distribution of the coefficients
2) $P(Y \mid X, b, \beta)$ is the likelihood term, representing the likelihood of the data given the coefficients.
3) $P(b \mid \alpha)$ is the prior distribution of the coefficients, controlled by the hyperparameter α.
4) β is the precision of the error term.

To find the posterior distribution, we can compute the mean of the posterior distribution using linear algebra. The posterior distribution is also a multivariate normal distribution.

The mean of the posterior distribution is given by:

$$\mu\_posterior = (X^T X + \alpha I)^{(-1)} X^T Y$$

Where

1) X^T is the transpose of the X matrix.
2) α is the precision hyperparameter for the prior distribution.
3) I is the identity matrix.

Thus we can predict the 'b' matrix by the above equation.

We know that :

Y = Xb + ε
Expected_Y = X(Expected_b) + Expected_ε(=0)
Expected_Y = X(Expected_b)

Thus we can predict the Expected value of Y of the testing dataset by the above equation.

Dataset used:
In order to illustrate the working of the Naive Bayes' classifier, we have used the advertising dataset that captures sales revenue generated with respect to advertisement spends across multiple channels like radio, tv and newspaper. We have taken 150 examples in the training dataset and 50 examples in the testing dataset.

Screenshot of some Training examples:

	TV	Radio	Newspaper	Sales
1	230.1	37.8	69.2	22.1
2	44.5	39.3	45.1	10.4

Final output and results:

1) **Comparison between Predicted and Actual Values**: The code computes predicted values for sales using the Bayesian linear regression model. It then compares these predicted values to the actual sales values from our dataset.

2) **Beta Value**: The code calculates the beta value, which represents the precision of the error term in your model.

   The value is calculated by finding the variance of (Y-Xb) = error term and reciprocating it.

3) **Root Mean Square Error (RMSE)**: The RMSE is a metric used to measure the accuracy of our model's predictions.

Screenshot of the Output:

```
Predicted Output Actual Output
18.68                  16.1
10.1397                11.6
16.4808                16.6
18.181                 19
15.6254                15.6
5.24513                3.2
15.2035                15.3
10.2393                10.1
10.1376                7.3
12.4659                12.9
14.3405                14.4
13.4261                13.3
15.0956                14.9
17.3187                18
11.1418                11.9
14.5553                11.9
10.5799                8
```

This is the screenshot of the output value of BETA and RMSE :-

```
Beta(PRECISION OF ERROR TERM): 0.349131
RMSE(ROOT MEAN SQUARE ERROR) : 1.63068
```