Bayesian Regression: Capstone Project Report

Student Name: Zafir Shaikh

Course: Machine Learning AI

Institution: Humber IGS

Course Code: BINF\_5507

Date: July 21, 2025

# 1. Introduction

Bayesian Regression is a probabilistic extension of linear regression that models uncertainty in predictions using probability distributions. It leverages Bayes’ Theorem to update prior beliefs with observed data, producing posterior distributions over model parameters rather than fixed values. This approach is particularly valuable when working with small datasets or when it is important to quantify uncertainty in predictions.  
  
Unlike classical regression, which gives a single estimate for each parameter, Bayesian Regression provides a full distribution that reflects the confidence in each estimate. This method is widely used in applications where understanding the confidence of predictions is critical, such as in healthcare diagnostics, financial forecasting, and scientific simulations.

# 2. How It Works

Bayesian Regression is grounded in Bayes’ Theorem:  
  
 P(θ | D) = (P(D | θ) \* P(θ)) / P(D)  
  
Where:  
- θ: Model parameters (e.g., coefficients)  
- D: Observed data  
- P(θ): Prior probability of the parameters  
- P(D | θ): Likelihood of data given the parameters  
- P(θ | D): Posterior distribution after seeing data  
  
Steps:  
1. Specify Priors: Prior beliefs about coefficients (usually Gaussian).  
2. Compute Likelihood: Probability of the observed data given model assumptions.  
3. Compute Posterior: Updated belief after observing data.  
4. Make Predictions: Use posterior to estimate outcome distributions.  
  
This results in a predictive distribution for new data, which includes both mean predictions and uncertainty intervals.

# 3. Use Cases & Performance

Applications:  
- Healthcare: Predicting patient risk levels with confidence bounds.  
- Finance: Modeling market trends while incorporating uncertainty.  
- Environmental Science: Simulating uncertain phenomena like climate changes.  
  
Strengths:  
- Provides confidence intervals for predictions.  
- Incorporates prior knowledge (useful with limited data).  
- Automatically penalizes overfitting through prior regularization.  
  
Limitations:  
- Computationally intensive, especially with large datasets.  
- Choice of priors can significantly influence results.  
- Interpretability can be more complex than traditional models.

# 4. Code Implementation Overview

Although no code is implemented in this report, here is how Bayesian Regression is commonly used:  
  
Tools:  
- scikit-learn: via BayesianRidge()  
- PyMC3 or Pyro: for full Bayesian probabilistic programming  
  
Typical Workflow:  
1. Load dataset  
2. Define prior distributions  
3. Train using a Bayesian regression class  
4. Extract posterior mean and intervals for predictions  
5. Compare performance with traditional models  
  
This makes Bayesian Regression suitable for pipelines that need interpretable uncertainty.

# 5. Comparison / Visualization

Feature Comparison Table:  
  
| Feature | Linear Regression | Bayesian Regression |  
|-----------------------------|--------------------------|--------------------------------|  
| Prediction Type | Point Estimate | Predictive Distribution |  
| Handles Uncertainty | No | Yes |  
| Incorporates Priors | No | Yes |  
| Overfitting Control | Needs manual regularization | Regularized via Priors |  
| Interpretability | High | Moderate (depends on priors)  
  
In Bayesian Regression, each prediction comes with a confidence interval, while linear regression offers no uncertainty quantification.  
  
Visuals like the plot below can demonstrate how Bayesian models include predictive bands:  
  
[Image Placeholder: Comparison of linear vs Bayesian predictions with confidence intervals]

# 6. References

1. Bishop, C. M. (2006). Pattern Recognition and Machine Learning. Springer.  
2. Murphy, K. P. (2012). Machine Learning: A Probabilistic Perspective. MIT Press.  
3. scikit-learn Documentation – https://scikit-learn.org  
4. PyMC3 Documentation – https://docs.pymc.io