UNIVERSITY MOHAMED VI
POLYTECHNIC
COLLEGE OF COMPUTING - QFM

Comparative Analysis of Stock Market Volatility Models: A Bayesian Regression vs. Linear Regression Approach

for final project of course:"Advanced Statistical Methods for Modeling and Finance - QFM1"

By
Généreux Akotenou

Supervised by
Pr. Ravi Prakash Ranjan

2023-2024

Contents

List of Figures

List of Tables

Abstract

In the dynamic realm of financial markets, understanding and predicting stock market volatility is a pivotal endeavor. Volatility, a measure of the degree of variation in stock prices, plays a crucial role in risk assessment, portfolio management, and investment decision-making. As we navigate the intricate landscape of financial analytics, our study endeavors to shed light on the nuanced nature of stock market volatility by employing two distinct yet powerful regression methodologies: Bayesian Regression and Linear Simple Regression. Our exploration is anchored in the empirical examination of Apple stock price data, sourced from Yahoo Finance. The dataset spans a critical period, from January 1, 2020, to December 1, 2023, encapsulating a spectrum of market events, economic shifts, and global dynamics. This chosen timeframe serves as the crucible for our comparative analysis, allowing us to capture the essence of market volatility and assess the efficacy of Bayesian and Linear Regression in this context.

Résumé

Dans le domaine dynamique des marchés financiers, comprendre et prédire la volatilité du marché boursier est très cruciale. La volatilité, mesure du degré de variation des prix des actions, joue un rôle essentiel dans l'évaluation des risques, la gestion de portefeuille et la prise de décision en matière d'investissement. Alors que nous naviguons dans le paysage complexe de l'analyse financière, notre étude s'efforce de mettre en lumière la nature nuancée de la volatilité du marché boursier en utilisant deux méthodologies de régression distinctes mais puissantes : la régression bayésienne et la régression linéaire simple. Notre exploration est ancrée dans l'examen empirique des données sur les cours de l'action d'Apple, collectées méticuleusement auprès de Yahoo Finance. Le jeu de données couvre une période critique, du 1er janvier 2020 au 1er décembre 2023, englobant un éventail d'événements de marché, de changements économiques et de dynamiques mondiales. Ce laps de temps choisi sert de creuset à notre analyse comparative, nous permettant de saisir l'essence de la volatilité du marché et d'évaluer l'efficacité des régressions bayésienne et linéaire dans ce contexte.

Introduction

Regression stands out as one of the most extensively employed statistical methods in applied sciences, serving the purpose of establishing connections between variables and predicting a target variable based on other explanatory variables, often referred to as features. One conventional approach involves the application of linear or polynomial regression, characterized as a frequentist method wherein each model parameter is treated as an unknown fixed point. Alternatively, Bayesian regression takes a distinctive perspective by considering the parameters themselves as random variables, rooted in Bayesian statistics. This approach introduces a probabilistic framework for modeling relationships between variables. Despite the inherent uncertainty within the Bayesian paradigm, its adaptability and incorporation of prior knowledge render it a potent tool. This becomes particularly valuable when dealing with sectors featuring prior information and highly volatile data, such as stock prices.

This paper embarks on a comprehensive exploration of stock price forecasting, beginning with an in-depth examination of linear regression intricacies. Subsequently, Bayesian regression is introduced as a sophisticated tool adept at incorporating prior information. The practical application of both methods is then showcased through a dedicated case study session, elucidating the nuances of each approach. To assess their performance, quantitative metrics such as Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) are employed, furnishing a robust foundation for comparison.

1 Prerequisites

In this first chapter, we discuss the prerequisites, starting with an understanding of what simple linear regression is. We delve into the Bayesian approach to regression, and finally, we explore the evaluation metrics for estimations made by regression models.

1.1 Frequentist Simple Linear Regression

The statistical technique widely utilized for analyzing multidimensional data is the linear regression model. In this section, we shortly present frequentist Simple Linear Regression.

1.1.1 Model

The principle of this model is to assume that a variable Y is explained, and modeled by a function of a single explanatory variable X. The simple linear regression model relies on expressing each observation y_i and is defined by the following formula:

$$y_i = b_0 + b_1 x_i + \varepsilon_i, \quad i = 1, \ldots, n$$

where:

- y_i : the i-th observation of the random variable to be explained Y
- x_i represents the i-th observation for the explanatory variable,
- ϵ_i is the error,
- The model parameters b_0 and b_1 are unknown constants.

Note that the multiple linear regression model can be expressed in matrix form as:

$$\mathbf{Y} = \mathbf{X}\beta + \epsilon,$$

where:

- $\mathbf{Y}$ is a random vector of dimension n,
- The data is arranged in a matrix $\mathbf{X}$ of dimension $(n, p+1)$ in $R^{n\times(p+1)}$, with the first column containing the unit vector indicating the constant b_0 in the equation,
- β is the parameter vector of the model with dimension $p+1$,
- ϵ is the error vector of dimension n.

1.1.2 Model Hypotheses

Here, we assume the error ϵ_i is independent and identically distributed as normal random variables with mean zero and constant variance σ^2:

$$\epsilon_i \sim \text{Normal}(0, \sigma^2)$$

1.1.3 Estimation of parameters

To estimate the parameters of the model, one can use the ordinary least squares (OLS) method, which does not require any additional assumption about the distribution of . Alternatively, one can use the Maximum Likelihood (ML) method, which is based on the normality of . In the subsection we will just explor the first approach.

We denote the Ordinary Least Squares (OLS) estimators of b_0 and b_1 as $\hat{b}_0$ and $\hat{b}_1$, obtained by minimizing the quantity:

$$\gamma(b_0, b_1) = \sum_{i=1}^{n} \epsilon^2 = \sum_{i=1}^{n} (y_i - (\hat{b}_0 + \hat{b}_1 x_i))^2$$

To get the optimal parameters we have to derivate γ and extract parameters values when the derivatives is equal to zero.

$$\left\{ \begin{array}{l|l} \frac{\partial\gamma(b_0,b_1)}{\partial b_0} = 0, & b_0 = \hat{b}_0 \\ \frac{\partial\gamma(b_0,b_1)}{\partial b_1} = 0, & b_1 = \hat{b}_1 \end{array} \right.$$

Let's $\bar{x}$ and $\bar{y}$ are the empirical means of x_i and y_i (respectively), S_x and S_xy be defined as follow:

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i \quad \text{and} \quad \bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$$

$$S_x = \sum_{i=1}^{n} (x_i - \bar{x})^2 = \sum_{i=1}^{n} (x_i^2 - n\bar{x}^2)$$

$$S_{xy} = \sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y}) = \sum_{i=1}^{n} (x_i y_i - n\bar{x}\bar{y})$$

After some calculations, we deduce:

$$\left\{ \begin{array}{l} \hat{b}_0 = \bar{y} - \hat{b}_1\bar{x} \\ \hat{b}_1 = \frac{S_{xy}}{S_x} \end{array} \right.$$

Under the normallity of ϵ, the estimators $\hat{b}_0$ and $\hat{b}_1$ are unbiased.

1.2 Bayesian Linear Regression

In this chapter, we now turn to the Bayesian version of linear regression and show that under the reference prior, we will obtain the posterior distributions of α and $beta$ analogous with the frequentist OLS results.

1.2.1 Definition

In Bayesian Regression, several key terms play a crucial role in shaping the model and understanding the underlying probabilistic framework.

- **Prior Distribution ($p(\theta)$):** The initial belief about the distribution of parameters (θ) before observing any data.

- **Likelihood ($p(\mathbf{y}|\mathbf{X}, \theta)$):** The probability of observing the data ($\mathbf{y}$) given the parameters (θ) and the predictor variables ($\mathbf{X}$).

- **Posterior Distribution ($p(\theta|\mathbf{X}, \mathbf{y})$):** The updated distribution of parameters (θ) after incorporating the observed data. It combines the prior beliefs and likelihood.

- **Evidence or Marginal Likelihood ($p(\mathbf{y}|\mathbf{X})$):** The probability of observing the data ($\mathbf{y}$) without considering specific values for the parameters. It acts as a normalization constant in Bayesian inference.

- **Hyperparameters:** Parameters that define the distribution of the prior.

1.2.2 Goal

In the Bayesian framework, our objective is to refine the probability distributions governing the unknown parameters b_0, b_1, and σ^2. This refinement is achieved through an update process that incorporates observed data. The Bayesian approach enables us to iteratively enhance our understanding of the posterior distributions for the regression intercept (b_0), slope (b_1), and the variance of the error term (σ^2), leveraging the available dataset to inform and update our prior beliefs.

1.2.3 The standard linear Model

The Bayesian model starts with the same model as the classical frequentist approach:

$$y_i = b_0 + b_1 x_i + \varepsilon_i, \quad i = 1, \ldots, n$$

under the assumption that the errors ϵ are independent and identically distributed as normal random variables with mean zero and constant variance σ^2. This assumption is exactly the same as in the classical inference case for testing and constructing confidence intervals for different parameters.

Under the assumption that the errors ϵ_i are normally distributed with constant variance σ^2, we have for the random variable of each response Y_i, conditioning on the observed data x_i and the parameters b_0, b_1, σ^2, is normally distributed:

$$Y_i | x_i, b_0, b_1, \sigma^2 \sim \text{Normal}(b_0 + b_1 x_i, \sigma^2), \quad i = 1, \ldots, n.$$

The likelihood of Y is given by:

$$\mathcal{L}(b_0, b_1, \sigma) = \prod \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(y_i - (\alpha + \beta x_i))^2}{2\sigma^2}\right)$$

- Modeling
  Let use general equation for multiple variable feature models. The equation is given by:

  $$f(x) = X^T W \text{ and } y = f(x) + \varepsilon$$

  Where x is the input vector, w is a vector of parameters of the linear bias, f is the function value, and y is the observed target value. We have assumed that the observed values y differ from the function values $f(x)$ by additive noise, and we will further assume that this noise follows an independent, identically distributed Gaussian distribution with zero mean and variance σ_n^2. The noise term ε follows a normal distribution with zero mean and variance σ_n^2, i.e., $\varepsilon \sim \mathcal{N}(0, \sigma_n^2)$. We infer that :

  $$Y \sim \mathcal{N}(f(x) = X^T W, \sigma_n^2 I)$$

- Prior
  In the Bayesian framework, it is essential to define a prior distribution over the parameters, representing our beliefs about these parameters before observing any data. In this context, we select a zero mean Gaussian prior with covariance matrix Σ_p on the weigh. Let's say that the weight vector w follows a normal distribution with zero mean and covariance matrix Σ_p, i.e., $w \sim \mathcal{N}(0, \Sigma_p)$.

- Bayes Rules for Posterior distribution
  In Bayesian linear model inference, we derive conclusions from the posterior distribution over the weights. This distribution is computed using Bayes' rule, as follows:
  $$\text{posterior} = \frac{\text{likelihood} \times \text{prior}}{\text{marginal likelihood}},$$

  The posterior weight distribution is defined as:
  $$p(w|y,X) = \frac{p(y|X,w)p(w)}{p(y|X)}.$$

  where the normalizing constant, also known as the marginal likelihood, is independent of the weights and given by:
  $$p(y|X) = \int p(y|X,w) \cdot p(w) dw$$

  By neglecting the normalization constant, we obtain an approximation of the posterior distribution as follows:
  $$\begin{aligned} p(w|y,X) &\approx p(y|X,w)p(w) \\ &\approx exp(-\frac{1}{2\sigma_n^2}(y - X^T w)^T (y - X^T w)) exp(-\frac{1}{2} w^T \Sigma^{-1} w) \\ &\approx exp(-\frac{1}{2}(w - \bar{w})^T (\frac{1}{\sigma_n^2} X X^T + \Sigma_p^{-1})(w - \bar{w})) \\ &\approx \mathcal{N}(\bar{w} = \frac{1}{\sigma_n^2} A^{-1} X y, A^{-1}), \text{ where } A = \frac{1}{\sigma_n^2} X X^T + \Sigma_p^{-1} \end{aligned}$$

- Prediction

  - Prediction distribution:
    To make predictions for a test case we average over all possible parameter values, weighted by their posterior probability:
    $$p(f^*|x^*, X, y) = \int p(f^*|x^*, w) p(w|X, y)\, dw = \mathcal{N}\left(\frac{1}{\sigma_n^2} x^* A^{-1} X y, x^* A^{-1} x^*\right)$$
  - Sample from the Posterior Distribution:
    We will draw samples from the posterior distribution of the parameters. For each set of parameters, we generate predictions for the target variable y based on the input features.
  - Aggregate Predictions:
    Finally, we will collect the predictions generated from each sample of parameters.

1.3 Evaluation Metric

When assessing the performance of a predictive model, it is essential to use appropriate evaluation metrics. Two commonly used metrics for regression models are the Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE).

1.3.1 Mean Absolute Error (MAE)

The Mean Absolute Error is calculated as the average absolute difference between the predicted values and the actual values. It is defined as:

$$MAE = \frac{1}{n} \sum_{i=1}^{n} |y_i - \hat{y}_i|$$

where n is the number of observations, y_i is the actual value, and $\hat{y}_i$ is the predicted value. MAE provides a measure of the average magnitude of errors, with lower values indicating better performance.

1.3.2 Root Mean Squared Error (RMSE)

The Root Mean Squared Error is another widely used metric that penalizes larger errors more heavily than MAE. It is calculated as the square root of the mean of squared differences between predicted and actual values:

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2}$$

RMSE indicates the typical size of errors and is sensitive to outliers. Like MAE, lower RMSE values correspond to better model performance. When comparing models or assessing the effectiveness of a regression model, both MAE and RMSE can provide valuable insights into the accuracy of predictions.

2 Application on Apple Stock prizes

In this final section, we tested our two prediction approaches, namely simple linear regression and Bayesian regression, on Apple's data to forecast stock prices. We elaborate on our methodology for data collection, analysis, and prediction, as well as provide detailed results and their interpretations.

2.1 Data collection and processing

To apply our predictive model on Apple stock prices, we start by collecting historical data using the Yahoo Finance API. Specifically, we retrieve historical stock price data for Apple (AAPL) from January 1, 2020, to December 30, 2023, using the following code:

2.1.1 Dataset Overview

The dataset includes the following columns: `Open`, `High`, `Low`, `Close`, `Volume`, `Dividends`, and `Stock Splits`. Each column represents specific aspects of Apple stock pricing and financial information:

- `Open`: The opening price of the stock on a given day.
- `High`: The highest price the stock reached during the trading day.
- `Low`: The lowest price the stock reached during the trading day.
- `Close`: The closing price of the stock on a given day.
- `Volume`: The total number of shares traded during the day.
- `Dividends`: Dividend payments made to shareholders on the corresponding day.
- `Stock Splits`: Any stock split that occurred on the given day.

Image 1 shows APPLE stock prices evolution in the targeted period.

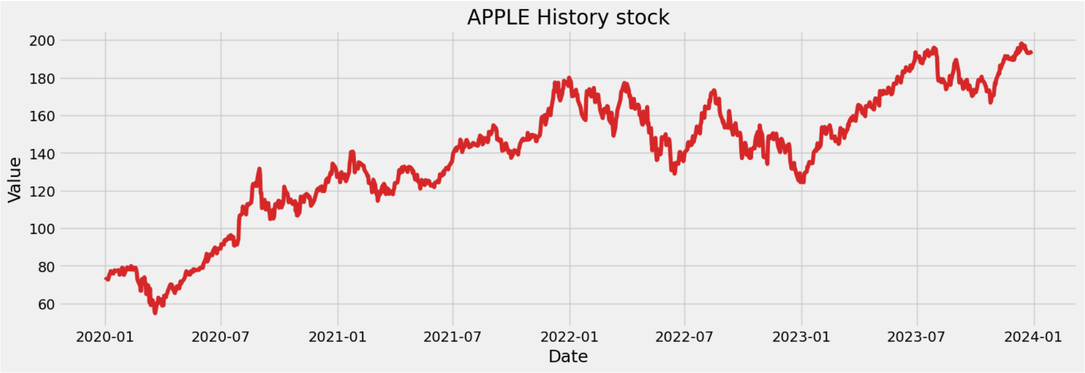


Figure 1: APPL price evolution in the target period

2.1.2 Correlation Test

Our objective is to predict tomorrow's stock price based on various features. To achieve this, we duplicate the 'Close' column and shift it, making the next day's 'Close' value our target variable, which we denote as 'FutureClose' for a given day.

In Figure 2, we observe the correlation analysis between different features and the 'FutureClose.' Our focus is on understanding how well the 'Close' prices and 'Volume'

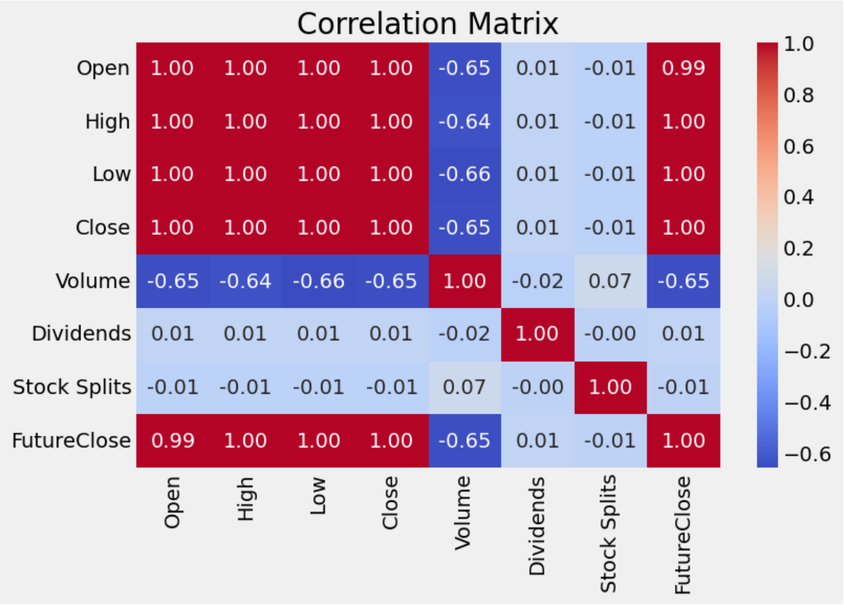


Figure 2: Correlation Analysis between Features and FutureClose

correlate with the future stock prices. Based on this analysis, we choose to retain 'Close' prices and 'Volume' as our selected features for further modeling.

The table 1 presents key features, including 'Date,' 'Close' prices, 'Volume,' and 'FutureClose,' providing a snapshot of the data used for further modeling.

Date	Close	Volume	FutureClose
2020-01-02 00:00:00-05:00	73.15264129638672	135480400	72.44146728515625
2020-01-03 00:00:00-05:00	72.44146728515625	146322800	73.0186996459961
2020-01-06 00:00:00-05:00	73.0186996459961	118387200	72.6752700805664
2020-01-07 00:00:00-05:00	72.6752700805664	108872000	73.84432983398438
2020-01-08 00:00:00-05:00	73.84432983398438	132079200	75.41284942626953

Table 1: Overview of the final dataset after analysis.

2.1.3 Data Separation for Training and Testing

The process of training and testing a predictive model involves dividing the available dataset into two subsets: one for training the model and another for evaluating its performance. To ensure an effective evaluation of the model's generalization ability, we designate a specific period as the testing set. In our case, the testing set comprises the most recent 59 days of available data, with the testing end date calculated as the maximum date in the dataset (`test_end_date`). The start date (`test_start_date`) is determined by subtracting 59 days from the testing end date.

The training set, on the other hand, includes all data points leading up to the testing start date. This ensures that the model is trained on historical data, providing it with a foundation to make predictions based on patterns observed during this period.

2.2 Benchmarking the Regression Model

In assessing the performance of our regression model, we employ two standard metrics to gauge its accuracy and effectiveness. For this regression task, we rely on the following metrics:

- **Mean Absolute Error (MAE):** This metric calculates the average of the absolute differences between the predictions and the true values. It provides a measure of the average magnitude of errors.
- **Root Mean Squared Error (RMSE):** The square root of the average of the squared differences between the predictions and the true values. RMSE is particularly useful as it penalizes larger errors more significantly.

To establish a baseline for comparison, we implement a naive approach known as the "Median Baseline." In this approach, we predict the median value from the training set for all testing cases. This serves as a straightforward benchmark against which our sophisticated two-linear regression model will be compared. The result of the Median Baseline is as follows:

- **Median Baseline MAE:** 46.52
- **Median Baseline RMSE:** 46.89

If our machine learning model cannot surpass the performance of this simple baseline, it prompts us to explore alternative approaches and refine our model further.

2.3 Tools used

The implementation of the Linear Regression model and Bayesian Regression involves several Python libraries and tools for data manipulation, visualization, and statistical modeling. Below are the key tools utilized in this analysis:

- **Pandas and NumPy:** These libraries are employed for efficient data manipulation and numerical computations.
- **Matplotlib and Seaborn:** Matplotlib and Seaborn are used for creating informative plots and visualizations, aiding in the interpretation of the data.
- **Arviz:** Arviz is a library used for Bayesian data analysis and visualization. It provides tools for summarizing and visualizing Bayesian models.
- **yfinance:** This library facilitates the extraction of financial data from Yahoo Finance, simplifying the process of obtaining historical stock prices.
- **Scikit-Learn:** Scikit-Learn is utilized for splitting the data into training and testing sets and for implementing the Linear Regression model. It also provides tools for scaling features.
- **PyMC3:** PyMC3 is employed for Bayesian Inference. It allows us to specify and fit Bayesian models, providing a probabilistic framework for regression analysis.
- **DateTime:** The DateTime module is used for handling date and time data, essential for time-series analysis and model evaluation.

2.4 Prediction with Simple Linear regression

In this section, we delve into the application of Simple Linear Regression for predicting future stock prices. Utilizing the tools mentioned earlier, we train the model and extract essential parameters to understand how it interprets the relationship between the selected features and the target variable.

2.4.1 Model Evaluation: Parameters and metrics

After fitting the model, we obtain the regression coefficients and intercept:

- **Regression Coefficients:** [9.93456846e-01, -1.71035140e-09]
- **Regression Intercept:** 1.1715065100230788

These coefficients represent the weights assigned to the 'Close' prices and 'Volume' features, indicating their influence on the predicted 'FutureClose' values. Next, we use the trained model to make predictions on the test data. The performance of the Simple Linear Regression model is evaluated using the following metrics:

Index	**MAE**	**RMSE**
Baseline	46.516032	46.892033
Linear Regression	1.476316	1.851433

Table 2: Performance Metrics for Simple Linear Regression Model

2.4.2 Model Evaluation: Scatter Plots

To visually assess the performance of the Simple Linear Regression model, we present two scatter plots that compare the predicted values with the actual values. Figure 3 illustrates the scatter plot where each point represents a pair of actual and predicted 'FutureClose' values. This allows us to observe the alignment between the model's predictions and the ground truth. In Figure 3, the points closer to the diagonal line indicate accurate predictions, while deviations from the line represent prediction errors.

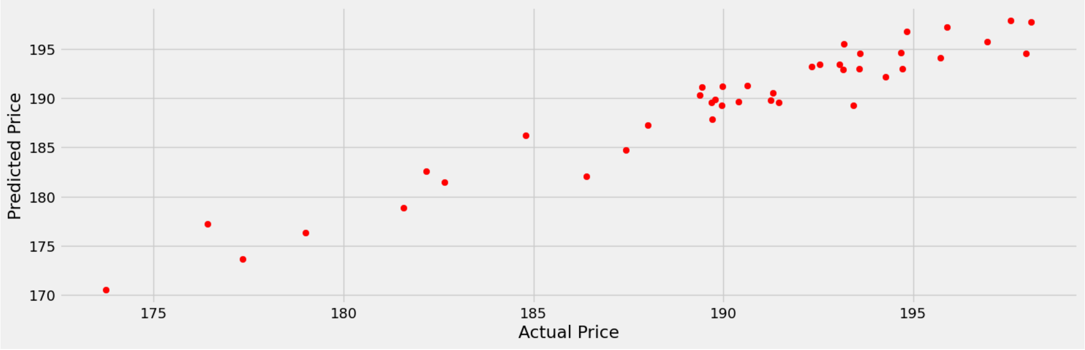


Figure 3: Scatter Plot of Predicted vs. Actual 'FutureClose' Values

To further delve into the comparison, 4 provides a visual representation of the real 'FutureClose' values in gray alongside the model's predictions in blue. This graph offers a more comprehensive view of how well the model captures the underlying trends in stock prices.

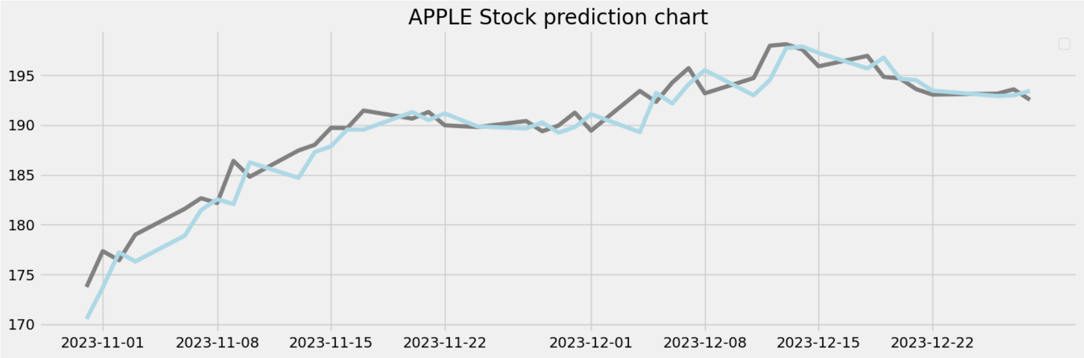


Figure 4: Real vs. Predicted 'FutureClose' Prices

2.5 Prediction with Bayesian Linear regression

In this section, we discuss Bayesian inference for our linear regression model.

2.5.1 Model

Let x_1 be the close price of the previous day x_2 be the total volume of the previous day and y_i be the closing stock price of the next day—our target variable that we aim to predict. The Bayesian model starts with the same model as the classical frequentist approach:

$$y_i = b_0 + b_1 x_1 + b_2 x_2 + \varepsilon_i, \quad i = 1, \ldots, n$$

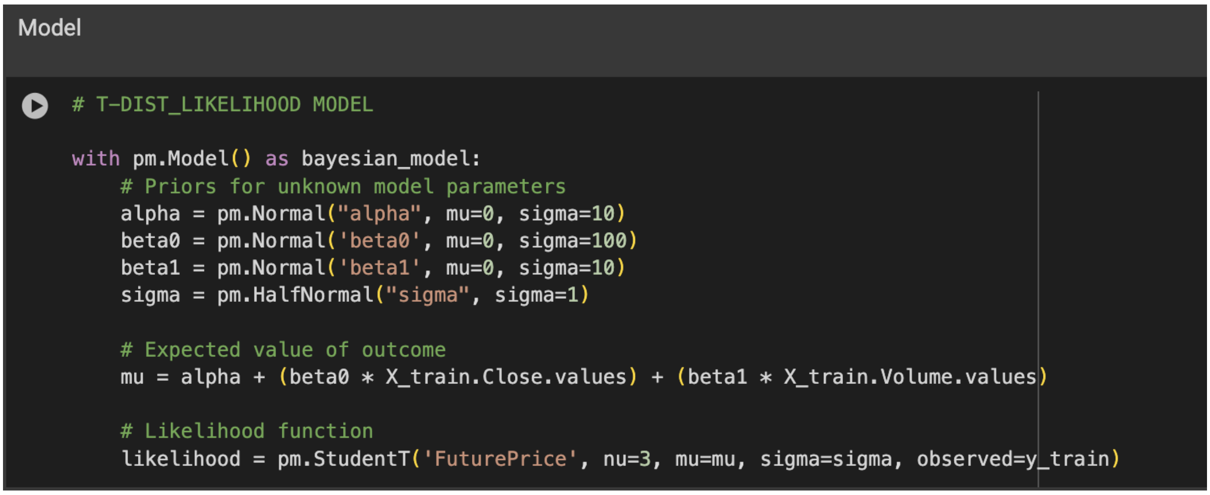
2.5.2 Prior Distributions

- We assume that the error term ϵ_i is independent and identically distributed according to the normal distribution $\epsilon_i \sim \text{Normal}(0, \sigma^2)$, where σ^2 represents the common variance shared among all observations.

- We also assume that the parameters b_0, b_1, b_2 are normally distributed

2.5.3 Fitting the Model

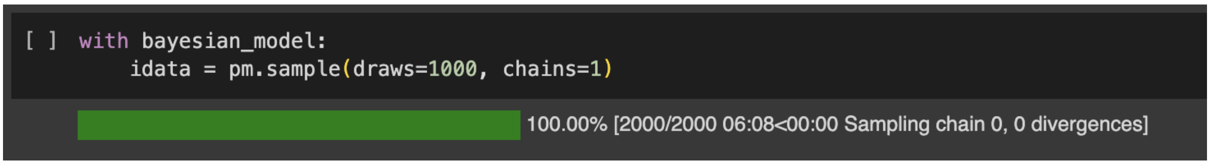
To construct the model, we employ the PyMC library to define the likelihood distribution and the prior distributions. Figure 5 illustrates the model composition using PyMC in Python language.



```
# T-DIST_LIKELIHOOD MODEL

with pm.Model() as bayesian_model:
    # Priors for unknown model parameters
    alpha = pm.Normal("alpha", mu=0, sigma=10)
    beta0 = pm.Normal('beta0', mu=0, sigma=100)
    beta1 = pm.Normal('beta1', mu=0, sigma=10)
    sigma = pm.HalfNormal("sigma", sigma=1)

    # Expected value of outcome
    mu = alpha + (beta0 * X_train.Close.values) + (beta1 * X_train.Volume.values)

    # Likelihood function
    likelihood = pm.StudentT('FuturePrice', nu=3, mu=mu, sigma=sigma, observed=y_train)
```

Figure 5: Illustration of the Bayesian regression model composition.

In this context, we employ the T-Student distribution due to the volatile nature of the data, and T-Student is particularly robust in handling outliers. Additionally, we impose a half-normal constraint on sigma to ensure that errors are restricted to positive values. Analytically deriving the posterior distribution can be challenging at times. Therefore, we utilize the MCMC (Markov Chain Monte Carlo) method to train the model and obtain the posterior distributions as illustrated in image 6



```
[ ]  with bayesian_model:
         idata = pm.sample(draws=1000, chains=1)
```

```
100.00% [2000/2000 06:08<00:00 Sampling chain 0, 0 divergences]
```

Figure 6: Training the model with Markov Chain Monte Carlo

The overview of the final model is described in Figure 7. The model image is automatically obtained from the PyMC model using model_to_graphviz function.

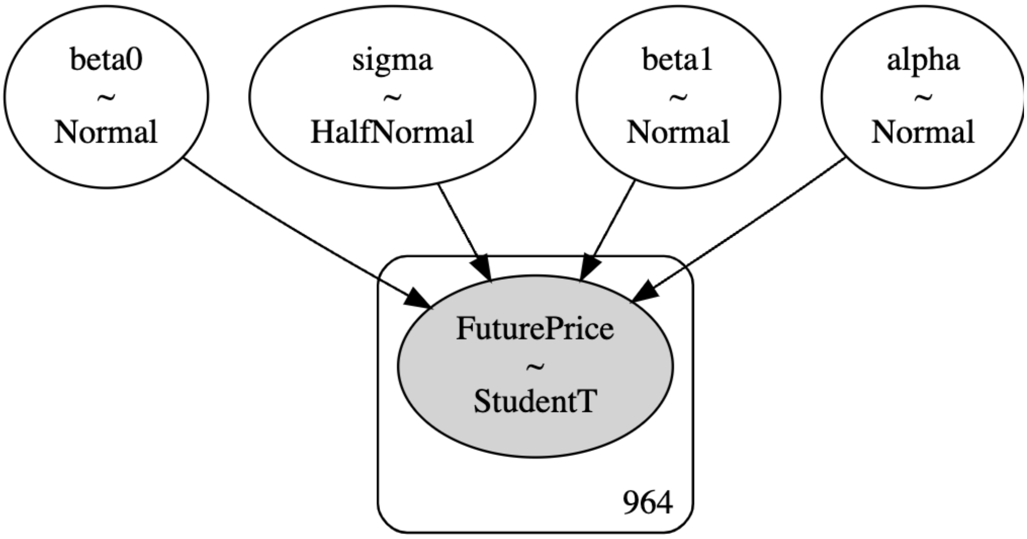


Figure 7: Bayesian model overview

2.5.4 Posterior Distributions

By leveraging the PyMC library, we obtain the posterior distributions of our parameters. This is visually represented in Figure 8. The table 3 provides a summary of the posterior distributions for each model parameter. The columns represent the mean, standard deviation (SD), 3% and 97% Highest Density Intervals (HDI), and the Standard Error of the Mean (MCSE) for each parameter.

Parameter	**Mean**	**SD**	**HDI_3%**	**HDI_97%**	**MCSE_Mean**	**MCSE_SD**
α	0.58	0.03	0.52	0.61	0.01	0.01
β_0	1.00	0.00	1.00	1.00	0.00	0.00
β_1	-0.00	0.00	-0.00	0.00	0.00	0.00
σ	1.89	0.06	1.79	2.01	0.01	0.00

Table 3: Summary statistics of the posterior distributions for each model parameter.

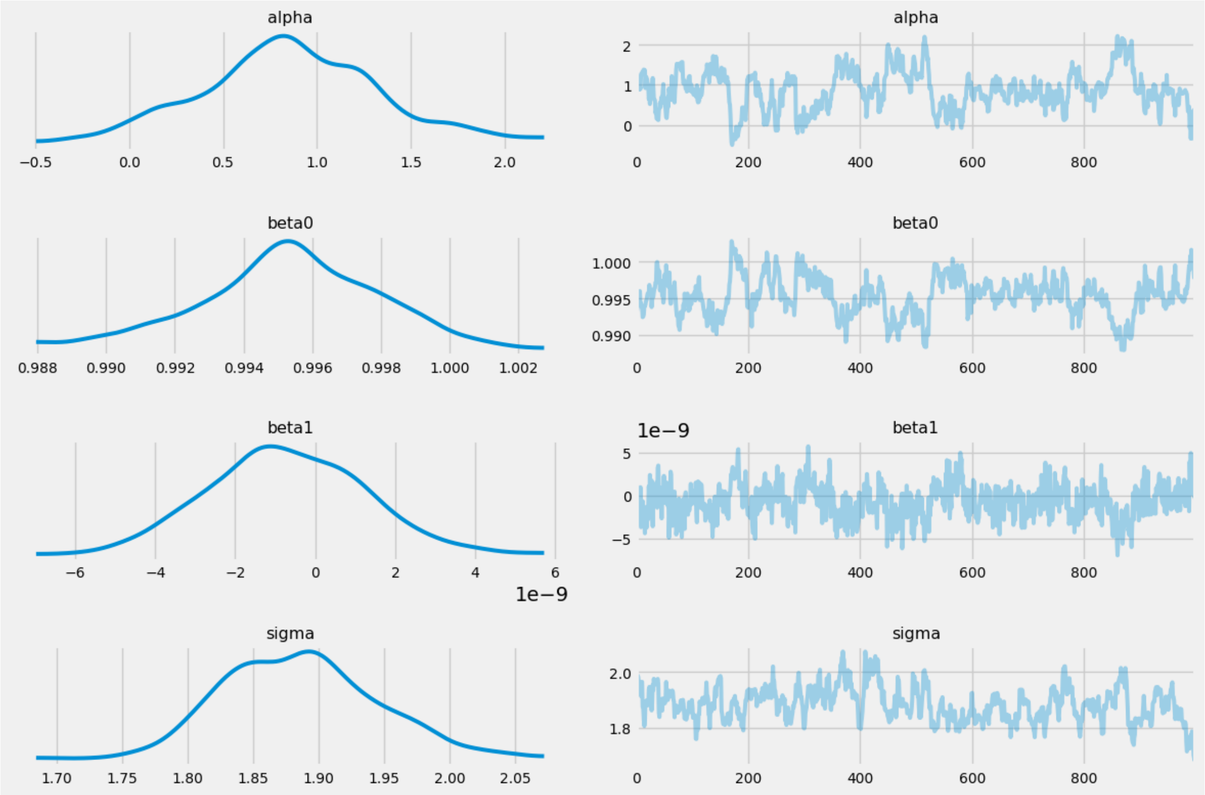


Figure 8: Model posterior distribution

2.5.5 Model Evaluation: Parameters and metrics

Next, we utilize the trained Bayesian Linear Regression model to make predictions on the test data. Considering that the posterior distribution means approximating the true parameters, the model's predictions align with those of a simple regression. The performance of the Bayesian Linear Regression model is then evaluated using the following metrics:

Index	**MAE**	**RMSE**
Baseline	46.516032	46.892033
Linear Regression	1.476316	1.851433
Bayesian Regression	1.45526	1.808408

Table 4: Performance Metrics for Bayesian Linear Regression Model

For a more in-depth comparison, we present a visual representation of the actual 'FutureClose' values in gray, juxtaposed with the Bayesian model's predictions in yellow and those of the simple linear regression model in blue. The graph in Figure 9 provides a comprehensive view of how effectively the models capture the underlying trends in stock prices.

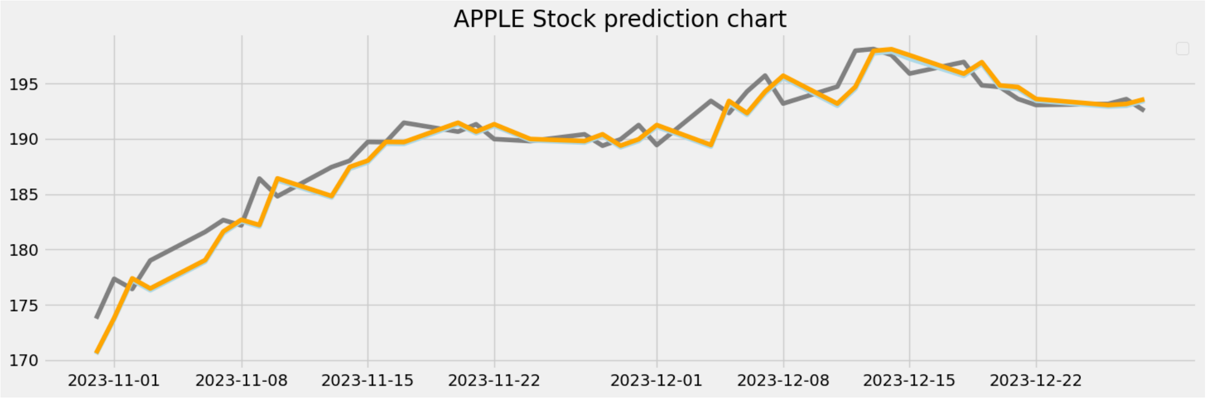


Figure 9: Bayesian Prediction vs. Real

Conclusion

In conclusion, our exploration into stock market prediction using Bayesian Linear Regression and Simple Linear Regression models has yielded valuable insights. The Bayesian approach allowed us to incorporate prior beliefs and uncertainty into the modeling process, resulting in posterior distributions that represent a more nuanced understanding of the parameter space.

The comparison between the Bayesian and simple regression models showcased their respective strengths and limitations. While the Bayesian model provides a richer characterization of uncertainty, the simple regression model offers a straightforward interpretation of parameter estimates. The evaluation metrics employed on the test data demonstrated the effectiveness of both models in predicting stock prices. Visual representations further illustrated the model performances, showcasing their ability to capture underlying trends. As we navigate the complex landscape of financial predictions, it is essential to consider the trade-offs between model complexity and interpretability. The Bayesian Linear Regression model, with its incorporation of uncertainty, stands as a powerful tool for modeling financial data.

References

1. Stats with R. "Introduction to Bayesian Regression."
2. Refrafi, Asma. "Régression Linéaire Bayésienne"
3. Yahoo Finance. "AAPL Historical Data."
4. PyMC. "Student's T Distribution."
5. PyMC: Model Bayesian with Python

Appendices

You can access the notebook used in this project along with the dataset files on the GitHub repository:

- GitHub Repository: https://github.com/Genereux-akotenou/Bayesian_regression