# Interest Rate Modeling

Rishabh Shah
*Rutgers University*
rishabhsunil.shah@rutgers.edu

Hemish Shah
*Rutgers University*
hemishmanish.shah@rutgers.edu

Aayush Patel
*Rutgers University*
aayush16.patel@rutgers.edu

## I. Abstract

This paper explores the development of a comprehensive interest rate model by employing object-oriented programming principles within Python. The model integrates multiple interest rate components, emphasizing modularity, flexibility, and maintainability, key aspects enabled by object-oriented design. We aim to develop and deploy a user-friendly interface for interacting with the model, enhancing accessibility for users who may not be well-versed in programming. The research focuses on single-factor short rate models, specifically the Merton Model, Vasicek Model, and Normal Model [1], which are pivotal in financial markets for the pricing of various fixed-income securities and derivatives. Each model offers unique insights into the dynamics of short-term interest rates, influenced by different sources of risk. Through this project, we document the implementation steps, from initial model setup to optimization and deployment, illustrating the practical applications and effectiveness of these models in predicting interest rate movements. The paper also discusses the potential for extending these methodologies to incorporate multi-factor models, thereby broadening the scope and accuracy of interest rate forecasting in complex financial environments.

## II. Introduction

Interest rate models are indispensable tools in the financial sector, providing essential insights for pricing securities, managing risk, and conducting monetary policy. They serve as the backbone for understanding and forecasting the behavior of interest rates over time, which is crucial for both financial institutions and policy makers. The accuracy of these models directly influences the profitability of investment strategies and the stability of financial markets. [2]

This project presents a comprehensive approach to modeling interest rates using object-oriented programming (OOP) principles in Python. OOP allows for creating models that are not only effective but also scalable and easy to maintain, an essential feature in the ever-evolving landscape of financial technologies. We focus on the development of a user-friendly interface that makes these complex models accessible to a broader audience, including those without deep programming expertise.

Our project centers on three primary models, each representing different aspects and dynamics of interest rates. The **Merton Model** [3] is a continuous-time model that posits the underlying asset prices follow a geometric Brownian motion, described by the equation:

$$r_{t+\Delta t} = r_t + \alpha \Delta t + \sigma \sqrt{\Delta t} \cdot \epsilon$$

where $r_t$ is the interest rate at time $t$, $\alpha$ represents the drift or trend of the rate changes, $\sigma$ denotes the volatility, and $\epsilon$ is a standard normal random variable, capturing the stochastic nature of rate fluctuations.

The **Vasicek Model** [4], a cornerstone in financial mathematics, assumes that the evolution of interest rates is governed by an Ornstein-Uhlenbeck process, leading to the equation:

$$\Delta r_{t+\Delta t} = (\alpha - \beta r_t)\Delta t + \sigma \sqrt{\Delta t} \cdot \epsilon$$

This equation highlights the mean-reverting nature of interest rates, where $\beta$ represents the speed of reversion.

The **Normal Model** (or Gaussian Short Rate Model) [5], which, unlike the log-normal assumptions in other models, assumes that the short rate has a normal distribution. The model is defined by:

$$r_{t+\Delta t} = r_t + \alpha(\mu - r_t)\Delta t + \sigma \epsilon \sqrt{\Delta t}$$

Here, $\mu$ represents the long-term mean rate, $\alpha$ the speed of mean reversion, and $\sigma$ the volatility, with $\epsilon$ once again modeling random shocks to the rate.

Each of these models offers unique benefits and is suited to specific types of financial instruments and market conditions. Our project elaborates on the implementation of these models using OOP principles, detailing the methods and classes designed to encapsulate the functionalities of each model, thereby providing a robust framework for interest rate prediction and analysis.

## III. Dataset

The dataset consists of time-series data of interest rates from various tenors ranging from 0.25 years to 10 years, both for Constant Maturity Swap (CMS) rates and Constant Maturity Treasury (CMT) rates.

TABLE I
Constant Maturity Swap (CMS) Rates

| Date | cms0.25 | cms2 | cms3 | cms5 | cms7 | cms10 |
|---|---|---|---|---|---|---|
| 1988-01-08 | 7.375 | 8.532 | 8.952 | 9.376 | 9.693 | 9.874 |
| 1988-01-15 | 7.250 | 8.229 | 8.613 | 8.975 | 9.278 | 9.461 |
| 1988-01-22 | 7.125 | 8.128 | 8.509 | 8.831 | 9.128 | 9.324 |
| 1988-01-29 | 6.875 | 7.871 | 8.233 | 8.588 | 8.905 | 9.110 |
| 1988-02-05 | 6.813 | 7.794 | 8.016 | 8.466 | 8.781 | 8.959 |

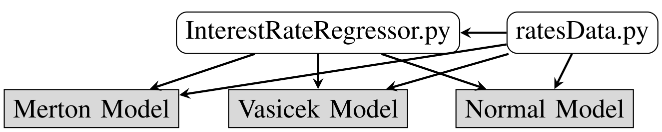
TABLE II
CONSTANT MATURITY TREASURY (CMT) RATES

| Date | cmt0.25 | cmt2 | cmt3 | cmt5 | cmt7 | cmt10 |
|---|---|---|---|---|---|---|
| 1988-01-08 | 6.75 | 7.92 | 8.16 | 8.48 | 8.80 | 8.97 |
| 1988-01-15 | 6.75 | 7.61 | 7.83 | 8.13 | 8.42 | 8.60 |
| 1988-01-22 | 6.70 | 7.51 | 7.73 | 8.01 | 8.29 | 8.49 |
| 1988-01-29 | 6.60 | 7.22 | 7.48 | 7.76 | 8.06 | 8.26 |
| 1988-02-05 | 6.40 | 7.10 | 7.32 | 7.64 | 7.95 | 8.12 |

Cleaning and Structuring: The data is cleaned to remove any null values and structured for analysis. Each row represents a date with corresponding interest rates across different maturities.
Normalization: Rates are normalized to ensure consistency across different time frames and to mitigate the impact of outlier values.

## IV. METHODOLOGY

The architecture of our interest rate modeling framework is designed to promote modularity, flexibility, and ease of maintenance, key aspects that are particularly critical in the dynamic field of financial modeling. This section provides a more detailed elaboration on the project's architecture, focusing on the structural organization, data management, computational models, and the planned user interface.




### A. Structural Organization

Our framework is structured around a core abstract base class named InterestRateRegressor. This class defines a blueprint for the necessary functionalities that any interest rate model within our framework must implement. These functionalities include computing discount factors, performing optimization, and predicting interest rates. Concrete implementations of this class are model-specific; for example, the Vasicek and Normal models are implemented as subclasses that inherit from InterestRateRegressor. This inheritance strategy not only enforces a consistent interface across different models but also facilitates the addition of new models without altering the existing system architecture.

### B. Data Management

Data handling is managed by the ratesData.py, which is crucial for preprocessing inputs to be fed into the interest rate models. This python file is responsible for loading data, parsing it according to the requirements of different models, and ensuring that it is correctly formatted. For instance, it handles tasks such as extracting relevant columns, sorting data based on time columns, and converting data types. The design of this module is such that it abstracts the complexities of data management from the user, allowing them to focus solely on modeling aspects.

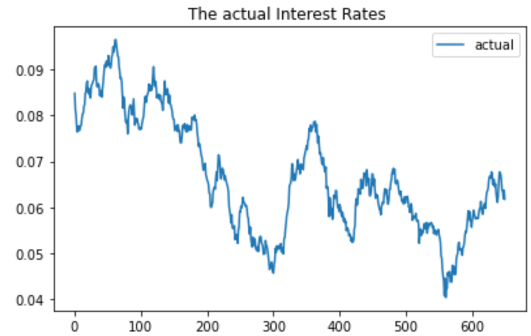



Fig. 1. Actual interest rates (CMT = 5)

### C. Model Implementation

Within our **ratesy** framework, the model implementation is encapsulated primarily within the **models.py** script, where each interest rate model is defined and implemented through its specific class. This structure not only allows for clarity and separation of concerns but also facilitates the extension and modification of individual models without impacting others. In this elaboration, we will focus on the implementations of the Vasicek, Merton, and Normal models, detailing the computational techniques and methodologies utilized for each.

*1) Vasicek Model Implementation:* The Vasicek Model Class is designed to implement the Vasicek interest rate model, which is a mean-reverting stochastic process [6]. This class is responsible for several key functionalities:
Initialization: It sets up the model with initial parameters, including the mean-reversion level (alpha), the speed of mean-reversion (beta), and the volatility of the rate changes (sigma). These parameters can be adjusted based on empirical data or optimization processes.
Parameter Estimation: This functionality involves calibrating the model parameters so that the model outputs closely match historical interest rate data. This usually requires an optimization algorithm that minimizes the difference between the model's predictions and actual historical rates.
Rate Calculation: The class includes a method to compute the short rate based on the Vasicek formula. This method calculates the next interest rate by considering the current rate, the parameters of the model, and a random term that introduces stochasticity, reflecting the inherent uncertainty in financial markets.
Simulation: The class provides a method for simulating future paths of interest rates. This is particularly useful for risk management and financial planning, allowing users to see potential future scenarios and prepare accordingly.

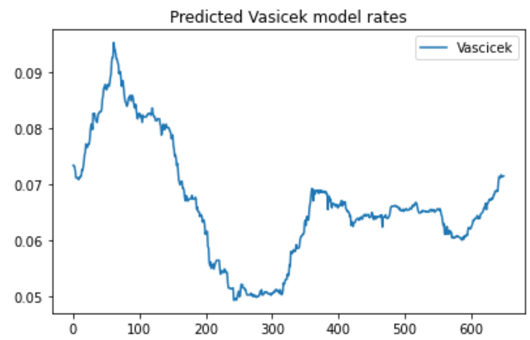



Fig. 2. Predicted Interest Rates using Vasicek Model

*2) Merton Model Implementation:* The Merton Model Class handles the implementation of the Merton model, primarily used for pricing options but adaptable for simulating the path of interest rates or other financial metrics under certain assumptions.
Initialization: Parameters such as drift and volatility, which are central to the geometric Brownian motion, are set up. These parameters define how the rate is expected to change on average (drift) and how volatile these changes are (volatility).
Random Walk Simulation: This class leverages the random walk hypothesis, assuming that changes in the metric being modeled (like interest rates) follow a log-normal distribution, facilitated by geometric Brownian motion. This involves generating random shocks based on a normal distribution, scaled by the model's volatility parameter.
Path Generation: It generates multiple potential future paths for the modeled financial metric. This functionality is crucial for creating scenarios that help in understanding the potential range of outcomes and the risks associated with various financial decisions.

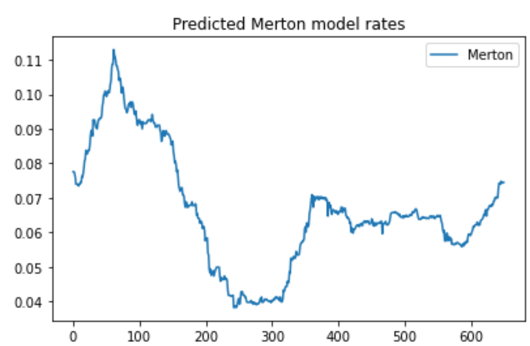



Fig. 3. Predicted Interest Rates using Merton Model

*3) Normal Model Implementation:* The Normal Model Class simplifies the simulation of interest rate changes by assuming that they are normally distributed, which is less complex than models requiring iterative calculations or complex tree structures.
Initialization: This involves setting up the normal distribution parameters, primarily the mean and standard deviation, which describe the expected change in interest rates and the variability around that change, respectively.
Direct Simulation: The class simulates changes in interest rates directly by drawing random samples from a normal distribution characterized by the specified mean and standard deviation. This method is efficient and straightforward, making it suitable for applications requiring a large number of simulations.
Calibration: The calibration method [7] adjusts the normal distribution parameters to better fit historical interest rate changes. This ensures that the simulations reflect realistic behavior observed in historical data, enhancing the model's predictive accuracy and reliability.

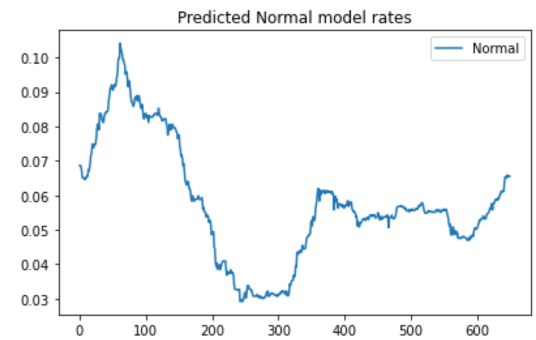



Fig. 4. Predicted Interest Rates using Merton Model

## V. RESULTS

In our project, we employed three distinct interest rate models—Vasicek, Merton, and Normal—to analyze and predict interest rates using historical data. The results of each model were evaluated based on their ability to fit and forecast 2-year interest rates, comparing them against actual historical rates.

The Vasicek model, renowned for its mean-reverting characteristics, was fitted to the historical interest rates, resulting in parameterized rates that were plotted alongside actual data. The outcomes demonstrated that the Vasicek model adheres closely to the underlying trends but may slightly deviate during periods of high volatility or market anomalies. [8]

TABLE III
PREDICTED INTEREST RATES USING THE VASICEK MODEL

| par_2.0 | par_3.0 | par_5.0 | par_7.0 | par_10.0 |
|---|---|---|---|---|
| 0.071120 | 0.072064 | 0.073354 | 0.074159 | 0.074891 |
| 0.071120 | 0.072064 | 0.073354 | 0.074159 | 0.074891 |
| 0.070696 | 0.071682 | 0.073036 | 0.073888 | 0.074665 |
| 0.069849 | 0.070919 | 0.072401 | 0.073345 | 0.074215 |
| 0.068157 | 0.069393 | 0.071133 | 0.072260 | 0.073316 |

Similarly, the Merton model, typically used for its robust handling of geometric Brownian motions in stock pricing, was adapted for interest rate predictions. The Merton model's results suggested a good fit, particularly capturing the peaks and troughs aligning with market expectations under normal market conditions.

TABLE IV
PREDICTED INTEREST RATES USING THE MERTON MODEL

| par_2.0 | par_3.0 | par_5.0 | par_7.0 | par_10.0 |
|---|---|---|---|---|
| 0.073196 | 0.074865 | 0.077590 | 0.079562 | 0.081294 |
| 0.073196 | 0.074865 | 0.077590 | 0.079562 | 0.081294 |
| 0.072678 | 0.074348 | 0.077074 | 0.079048 | 0.080782 |
| 0.071644 | 0.073313 | 0.076042 | 0.078019 | 0.079757 |
| 0.069575 | 0.071246 | 0.073980 | 0.075963 | 0.077709 |

The Normal model, characterized by its straightforward implementation, assuming normally distributed rate changes, provided a baseline simplicity in its predictions. Though less dynamic in capturing fluctuations, the Normal model offered a stable and consistent approximation of interest rates, beneficial for scenarios demanding less computational complexity.

TABLE V
PREDICTED INTEREST RATES USING THE NORMAL MODEL

| par_2.0 | par_3.0 | par_5.0 | par_7.0 | par_10.0 |
|---|---|---|---|---|
| 0.068652 | 0.068652 | 0.068652 | 0.068652 | 0.068652 |
| 0.068652 | 0.068652 | 0.068652 | 0.068652 | 0.068652 |
| 0.068135 | 0.068135 | 0.068135 | 0.068135 | 0.068135 |
| 0.067101 | 0.067101 | 0.067101 | 0.067101 | 0.067101 |
| 0.065035 | 0.065035 | 0.065035 | 0.065035 | 0.065035 |

Graphical representations of model performances revealed distinct patterns in how each model approached the prediction of interest rates, with the Vasicek and Merton models showing more responsiveness to changes in the actual rates compared to the Normal model. This comprehensive analysis not only underscores the strengths and limitations of each modeling approach but also enriches our understanding of applying different financial models to real-world data.

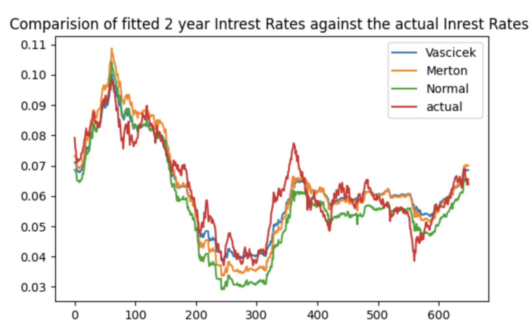



Fig. 5. Comparision of fitted 2 year Intrest Rates against the actual Inrest Rates

The performance of the Vasicek, Merton, and Normal models was quantified using the Root Mean Square Error (RMSE), a metric that reflects the average magnitude of the prediction errors. The Vasicek Model showed the highest accuracy with the lowest RMSE of 0.007029, indicating its strong ability to track actual interest rate movements. The Merton Model followed with an RMSE of 0.009737, effectively capturing the volatile swings in rates [9]. The Normal Model, with the highest RMSE of 0.013384, demonstrated a more generalized and less precise approach due to its simpler assumptions. These RMSE values not only illustrate each model's accuracy but also guide the selection of appropriate models based on specific analytical needs and data characteristics.

TABLE VI
RMSE VALUES FOR VARIOUS INTEREST RATE MODELS

| Model | RMSE |
|---|---|
| Vasicek Model | 0.007029 |
| Merton Model | 0.009737 |
| Normal Model | 0.013384 |

## VI. CONCLUSION

This research project effectively demonstrated the application of object-oriented programming principles in the development of interest rate models, emphasizing the utility of the Merton, Vasicek, and Normal models in financial analysis and prediction. Through Python, we crafted a user-friendly and adaptable framework for simulating and forecasting interest rate behaviors, crucial for financial planning and risk management.

The comparative analysis showed that each model has distinct advantages depending on market conditions and data characteristics. The Merton Model was adept at capturing log-normal movements of interest rates, while the Vasicek Model excelled in scenarios requiring predictions of mean-reverting behaviors. The Normal Model offered a straightforward and computationally efficient approach for basic rate change simulations.

The project underscores the critical role of structured data management and intuitive user interfaces in enhancing the accessibility and effectiveness of complex financial models. Each model's unique strengths highlight the importance of model selection in financial analysis, tailored to specific forecasting needs and market data characteristics.

## VII. FUTURE SCOPE

The development and deployment of a user-friendly interface (UI) represents a significant future enhancement for this project. This interface will allow users to interact directly with the interest rate models, enabling them to easily input data and receive forecasts. This will make the tool accessible to a broader audience, including those without deep technical expertise, thus democratizing the ability to predict interest rates for future investments. The UI will be designed to be intuitive and user-centric, ensuring that users can effortlessly navigate through different functionalities, from data input to visualization of model outputs.

Additionally, we plan to expand the capabilities of our modeling framework by incorporating multi-factor models. Specifically, the two-factor Vasicek model, which is renowned for its robustness and enhanced forecasting abilities, will be integrated into our suite of tools. This model is widely utilized in Wall Street for its ability to capture more complex dynamics of interest rates including parallel shifts and twists in the yield curve, providing a more comprehensive analysis and better forecasting. By incorporating such advanced models, our project aims to offer more precise and dynamic interest rate predictions, aligning with cutting-edge financial practices and catering to the sophisticated needs of modern financial markets.

These enhancements will not only improve the functionality and accuracy of our existing models but also significantly broaden the practical applications and appeal of our interest rate modeling framework.

## References


[1] Y. K. Kwok, *Mathematical models of financial derivatives*. Berlin: Springer-Verlag, 2008.

[2] D. Brigo and F. Mercurio, *Interest Rate Models - Theory and Practice: With Smile, Inflation and Credit*. Springer-Verlag Berlin Heidelberg, 2007.

[3] R. C. Merton, *Continuous-Time Finance*. Blackwell Publishers Inc., 1992.

[4] J. Halgašová, B. Stehlíková, and Z. Bučková, “Estimating the short rate from the term structures in the vasicek model,” *Tatra Mountains Mathematical Publications*, vol. 61, 12 2014.

[5] J. C. Hull, *Options, Futures, and Other Derivatives*, 8th ed. Prentice Hall, 2012.

[6] A. Halgasová, B. Stehlíková, and Z. Bučková, “Estimating the short rate from the term structures in the vasicek model,” *Tatra Mountains Mathematical Publications*, vol. 61, pp. 87–103, 2014.

[7] D. Sevčovič and A. Urbánová-Csajková, “Calibration of one factor interest rate models,” *Journal of Electrical Engineering*, vol. 55, pp. 46–50, 2004.

[8] O. Vasicek, “An equilibrium characterization of the term structure,” *Journal of Financial Economics*, vol. 5, pp. 177–188, 1977.

[9] R. J. Hyndman and G. Athanasopoulos, *Forecasting: principles and practice*, 2nd ed. OTexts, 2018.