Advanced Statistical Methods for Finance

Final Project done by

ADAMBI IDRISSOU Soulemane
&
Jean-Claude S. MITCHOZOUNOU

Africa Business School, UM6P, Morocco

Master of Quantitative and Financial Modelling (QFM)

Stock price simulation using Machine Learning and Monte Carlo

Teacher: Prof. Ravi Prakash Ranjan

Academic Year 2023-2024

University Mohammed VI Polytechnic (UM6P)

CONTENTS

Indroduction

The answer on the question, what will be the stock price in the specific time in the future is worth a fortune. Investors around the globe are seeking to know what the evolution of their stocks is in the future. Their motivation is to gain the maxim profit under the conditions of acceptable risk exposure and in the acceptable time range. These three basic attributes of the investments – return, risk and time – are three basic components of the investment triangle. Based on these attributes all investors make their decisions about strategic asset allocation. The requirements vary from investor to investor. The crucial decision factor is the risk aversion of the investor. Some investors are willing to accept higher level of risk, which is compensated with the higher return. On the other side, many investors are satisfied with lower gain, but more certain. The majority of the financial instruments follow random walk, so it is very difficult to predict the direction of their movements. In spite of that, there are some patterns, which can be generalized for some asset classes. This patterns help investor in their decision making processes.These models include machine learning and GBM models combined with Monte Carlo simulation.

Context

Monte Carlo simulation offers a powerful approach to anticipating future stock price fluctuations. This practical work focuses on applying this method to Microsoft to assess the potential variability of its stock prices.

Objective

The objectives set to answer the research questions include:
•To build a machine learning model based on past price histories to predict future share price using data analysis.
• To develop an approach based on the GBM model and Monte Carlo Simulation to extrapolate future stock prices.
• To compare and contrast the models and choose the best model.

A few reminders and an overviews

1.1 Defining Geometric Brownian Motion

[1, 2, 3, 4, 5]
Brownian motion (BM) was originally used to model the position of a particle in a fluid; a particle in a fluid is constantly colliding with other particles/molecules in the fluid, and the state is indeterminate. For similar reasons, Brownian motion is thought to be good modeling tool when model stock prices (here the state of interest is the price at a given time), because the price is being influenced by too many person to be determinant. The caveat is that the price cannot be negative, and stock prices usually follow exponential growth; thus, Geometric Brownian motion (GBM) is used, instead.

The definition of GBM is the solution to the following Stochastic Differential Equation (SDE):

$$dV(t) = \mu V(t)dt + \sigma V(t)dB(t), \tag{1.1}$$

where the differentials $dV(t)$, dt, and $dB(t)$ are the infinitesimal changes in price, time, and Brownian motion, respectively. Intuitively, the relationships described by the SDE make sense: the change in stock price at time t is some proportion μ of the current price plus some proportion σ of the current price scaled by some random variable that is related to Brownian motion.
$V(t)$ is a stochastic process and $B(t)$ a Wiener process, i.e

i) $B(0) = 0$;

ii) $\{B(t), t \geq 0\}$ has both stationary and independent increments;

iii) $\{B(t), t \geq 0\}$ has Gaussian increments (for $t > s > 0, [B(t) - B(s)]$ follows the Gaussian distribution)

iv) $\mathbb{E}(B(t)) = 0$

SDE in this study is that each increment in time results in the price of the stock moving with a drift $(\mu V(t)dt)$ and a shock $(\sigma V(t)dB(t))$. The *drift* can be seen as the general direction of the stock's price whereas the *shock* is a random amount of volatility that acts on the stock's price. The shock is what will create the curve's noise.

This SDE is not the best way to describe the evolution of stock prices, but it is the most widely accepted due to the complex nature in asset and stock prices. Solving the SDE requires Ito's calculus, and is non-trivial (See the wiki page for Geometric Brownian motion for more information). The analytical solution is

$$V(t) = v_0 e^{(\mu - \frac{1}{2}\sigma^2)t + \sigma B(t)} \tag{1.2}$$

Indeed, by posing $V_t = V(t), B_t = B(t)$ and applying Itô's lemma with $f(V_t) = \log(V_t)$ one has:

$$\begin{aligned} df &= f'(S_t)\,dV_t + \frac{1}{2}f''(V_t)(dV_t)^2 \\ &= \frac{1}{V_t}\,dS_t + \frac{1}{2}(-V_t^{-2})(V_t^2\sigma^2\,dt) \\ &= \frac{1}{V_t}\,(\sigma V_t\,dB_t + \mu V_t\,dt) - \frac{1}{2}\sigma^2\,dt \\ &= \sigma\,dB_t + \left(\mu - \tfrac{\sigma^2}{2}\right)\,dt. \end{aligned}$$

It follows that:
$\log(V_t) = \log(v_0) + \sigma B_t + \left(\mu - \frac{\sigma^2}{2}\right)t$,
exponentiating gives the expression for V :$V_t = v_0 \exp\left(\sigma B_t + \left(\mu - \frac{\sigma^2}{2}\right)t\right)$.

For our modeling purposes, $V(t)$ is the price of the stock (per share) at time t, and v_0 is the price at time $t = 0$ such that $V(0) = v_0$.

Expected Value:

To find the expected value of $V(t)$, we first examine a simpler case. Define Z to be a standard normal random variable $Z \sim N(0, 1)$. Let α be some scalar $\alpha \in (-\infty, \infty)$. Let's examine $\mathbb{E}[e^{\alpha Z}]$:

$$\mathbb{E}[e^{\alpha Z}] = \int_{-\infty}^{\infty} e^{\alpha x}\frac{1}{\sqrt{2\pi}}e^{-\frac{x^2}{2}}\,dx.$$

$$\begin{aligned} &= \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}}e^{\alpha x - \frac{x^2}{2}}\,dx \\ &= \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}}e^{\frac{\alpha^2}{2} - \frac{\alpha^2}{2} - \frac{x^2}{2} + \alpha x}\,dx \\ &= e^{\frac{\alpha^2}{2}}\int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}}e^{\alpha x - \frac{x^2}{2}}\,dx, \end{aligned}$$

where $\alpha x - \frac{\alpha^2}{2} - \frac{x^2}{2} = -\frac{1}{2}(x-\alpha)^2$. Then making the substitution, we get

$$\mathbb{E}[e^{\alpha Z}] = e^{\frac{\alpha^2}{2}}\int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}}e^{-\frac{1}{2}(x-\alpha)^2}\,dx,$$

where $\frac{1}{\sqrt{2\pi}}e^{-\frac{1}{2}(x-\alpha)^2}$ is the probability density function of a normal random variable (let's call it K) with mean α and variance 1: $K \sim N(\alpha, 1)$. Hence,

$$\int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}}e^{-\frac{1}{2}(x-\alpha)^2}\,dx = 1.$$

Then, the result is simply $\mathbb{E}[e^{\alpha Z}] = e^{\frac{\alpha^2}{2}}$. When dealing with Brownian motion, it is sometimes easier to think of $B(t)$ as a normal random variable with distribution $B(t) \sim N(0, t)$. Then, we can write $Z = \dfrac{B}{\sqrt{t}}$. Now, let's examine $\mathbb{E}[V(t)]$:

$$\begin{aligned} \mathbb{E}[V(t)] &= \mathbb{E}[v_0 e^{\mu t + \sigma B(t)}] \\ &= v_0 e^{\mu t}\mathbb{E}[e^{\sigma B(t)}] \\ &= v_0 e^{\mu t}\mathbb{E}[e^{\sigma\sqrt{t}Z}] \\ &= v_0 e^{\mu t}e^{\frac{1}{2}\sigma^2 t}. \end{aligned}$$

Therefore,

$$\mathbb{E}[V(t)] = v_0 e^{\mu t + \frac{1}{2}\sigma^2 t}.$$

Variance:

The variance of $V(t)$ is found similarly, by writing $\mathrm{Var}[V(t)]$ in terms of expected values:

$$\mathrm{Var}[V(t)] = \mathbb{E}[V(t)^2] - \mathbb{E}[V(t)]^2.$$

The second term is easily found to be:

$$\begin{aligned}\mathbb{E}[V(t)]^2 &= (v_0 e^{\mu t + \frac{1}{2}\sigma^2 t})^2 \\ &= v_0^2 e^{2\mu t + \sigma^2 t}.\end{aligned}$$

The first term is found in a way similar to which $\mathbb{E}[V(t)]$ was found:

$$\begin{aligned}\mathbb{E}[V(t)^2] &= \mathbb{E}[v_0^2 e^{2\mu t + 2\sigma B(t)}] \\ &= v_0^2 e^{2\mu t}\mathbb{E}[e^{2\sigma B(t)}] \\ &= v_0^2 e^{2\mu t}\mathbb{E}[e^{2\sigma \sqrt{t} Z}] \\ &= v_0^2 e^{2\mu t} e^{2\sigma^2 t}.\end{aligned}$$

By algebra, we get the following form:

$$\mathrm{Var}[V(t)] = v_0^2 e^{2\mu t + \sigma^2 t}(e^{\sigma^2 t} - 1).$$

1.1.1 Variables of the GBM model

a- **Drift**

Drift which is also known as the expected daily return of the stock is derived by calculating the mean, standard deviation, and variance according to a specific period of the historical performance of the stock . According to a report of Stock Price Predictions using GBM by Joel Liden, the formula of drift is:

$$Drift = \mu - \frac{1}{2}\sigma^2$$

Where μ is the mean of logarithmic returns of the stock and σ^2 is the variance.

b- **Shock**

According to a recent study[1], Malaysian gold prices were modelled using the GBM model. The study stated that the definition of Shock is the volatility of the asset price, which can be measured by calculating the standard deviation of the historical returns, given by the formula:

$$Shock = \sigma * Z(Rand(0;1))$$

Where σ is the standard deviation of the price and Z is a random number simulated following a normal distribution

1.2 Monte Carlo Simulation

[6, 7, 8, 9]
Monte Carlo Simulation can be used for predicting share price movements when the past share prices exhibit random behaviour without exhibiting high fluctuations. Monte Carlo experiment is a method to generate new sample from historical data. Monte Carlo Simulation in the study refers to a wide range of computational algorithms that utilize randomness in some way

to obtain a good approximation for probable outcomes.The main advantage of this method is then that replicated sample follows the same distributional properties as the original data. Once the variables of the GBM model are determined, Monte Carlo Simulations will be performed according to the formula:

$$\text{Price}_i = \text{Price}_{i-1} \cdot e^{\left(\mu - \frac{1}{2}\sigma^2\right) + \sigma \cdot Z(\text{Rand}(0,1))} \tag{1.3}$$

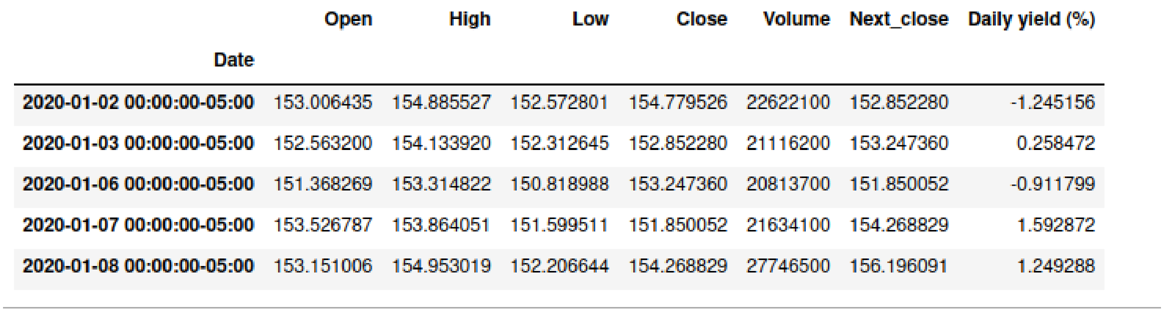
where Price_i is stock price today and Price_{i-1} is stock price yesterday.

The likelihood of the stock price following a given simulation is close to zero; therefore, Monte Carlo Simulations need to be repeated many times to determine the best fit line among all the simulations.

Data and Methodology

2.1 Data

The data used is historical Microsoft stock price data. The period considered extends from 2020-01-01 to 2023-10-01. This data was extracted using the yfinance Python library then processed and aggregated using the pandas and numpy libraries. We were interested in closing prices because they more closely summarize the state or variation of the stock price during the day. Each row of the Dataframe contains information regarding share price variations in a day as well as the volume of shares sold. To better analyze this data and calculate the parameters of the models to use, we have added two columns: the Next_close column which gives the closing price on the following day and the Daily yield (%) column which gives the daily yield as a percentage taking into account of the closing price on the previous day.



	Open	High	Low	Close	Volume	Next_close	Daily yield (%)
Date							
2020-01-02 00:00:00-05:00	153.006435	154.885527	152.572801	154.779526	22622100	152.852280	-1.245156
2020-01-03 00:00:00-05:00	152.563200	154.133920	152.312645	152.852280	21116200	153.247360	0.258472
2020-01-06 00:00:00-05:00	151.368269	153.314822	150.818988	153.247360	20813700	151.850052	-0.911799
2020-01-07 00:00:00-05:00	153.526787	153.864051	151.599511	151.850052	21634100	154.268829	1.592872
2020-01-08 00:00:00-05:00	153.151006	154.953019	152.206644	154.268829	27746500	156.196091	1.249288

Figure 2.1: DATA

Part of the data was used to determine the model parameters and the other part to test the model. For model validation we used a simple linear regression model. In this case, the data which was used to determine the parameters of the Brownian Model was this time used to train the Linear regression model and the rest of the data served as test data.

2.2 Model and Simulation

To model the evolution of Microsoft's stock prices, we chose to use the geometric Brownian motion model (GBM). This choice is motivated by the fact that the GBM is widely used in financial modeling to represent the stochastic behavior of stock prices. The GBM considers returns as a random variable following a normal distribution, which is consistent with the assumption that returns on financial assets are generally normally distributed.

After the choice of the model we used Monte Carlo Simulation for the simulation. Hence, we assumed the stocks prices evolution follows the equation (3) below. The specific parameters of this equation are determined as following:

- **Average Daily Return**

The average daily return was calculated by taking the logarithmic average of the daily percentage returns of Microsoft's stock prices. This gives us a measure of the daily average growth trend.

$$\mu = \ln\left(1 + \frac{1}{N}\sum_{i=1}^{N}\frac{P_i}{P_{i-1}}\right)$$

- **Daily Volatility (volatility)**

Daily volatility was measured as the standard deviation of daily returns, providing an indication of the daily variability of stock prices.

$$\sigma = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}\left(\ln\left(\frac{P_i}{P_{i-1}}\right) - \mu\right)^2}$$

The Monte Carlo simulation was performed by generating several possible stock price scenarios of Microsoft. Each scenario is based on the GBM model, which takes into account the initial price, the daily average return, and the daily volatility. For each iteration of the simulation, random returns are generated from a normal distribution and then used to update the simulated prices. This process is repeated to generate multiple scenarios, thereby providing a possible distribution of future stock prices for Microsoft. This approach allows us to explore the potential variability of Microsoft's stock prices by taking into account the specific characteristics of this company. In our case we also considered the average of the scenarios. This allowed us to make predictions taking into account all the scenarios generated by the Monte Carlo simulation.

After Monte Carlo simulation we used a Linear Regression model to do the comparison between predictions. Remember that our goal is to predict the stock price. Thus, the question formulated is the following: knowing that we have the information (price variation, volume sold, etc.) on the stocks on day n, what can be the closing price of the stocks on day n+1. Given the correlation between the variables of the collected data(Table 1), we decided to use as explanatory variables: the closing price and the sales volume of day n.

	Open	*High*	*Low*	*Close*	*Volume*	*Next_close*
Open	1.000000	0.998844	0.998769	0.997171	−0.343700	0.993409
High	0.998844	1.000000	0.998445	0.998646	−0.326870	0.994699
Low	0.998769	0.998445	1.000000	0.998711	−0.363338	0.994800
Close	0.997171	0.998646	0.998711	1.000000	−0.346846	0.995486
Volume	−0.343700	−0.326870	−0.363338	−0.346846	1.000000	−0.344967
Next_close	0.993409	0.994699	0.994800	0.995486	−0.344967	1.000000

Table 1: Correlation between the variables

RESULTS AND DISCUSSION

3.1 Plots and Discussions

Monte Carlo analysis was performed using Python and a large number of simulations to avoid distorted results. GBM model parameters such as drift and shock values were based on market data characteristics from 01/01/2018 to 05/08/2023. The Monte Carlo analysis was then incorporated into the GBM model data outputs to simulate the future evolution of market indices based on historical developments volatilities. Based on probability theory, the Monte Carlo technique is widely used and recommended to include uncertainties and typically, 1,000 or 10,000 analyzes are performed. In our study, the Monte Carlo Simulation is performed on Microsoft for a sample size. The overview of the stock and possible outcomes of the future is acquired by running multiple simulations, creating lots of random price curves, all different, but at least at the same time share some of the key features of the historical context price data. Too many executions can take time therefore in this research, 100 cycles are carried out since 08/05/2023. Based on Figure 2, it can be concluded that Microsoft's stock market performance remained stable or downward trend, as most sample size simulations end up closing lower than the opening price on 05/08/2023, while only one few simulations have reached new heights. This seems to indicate that it is possible to model the price of a stock using Monte Carlo simulations based on the GBM model.

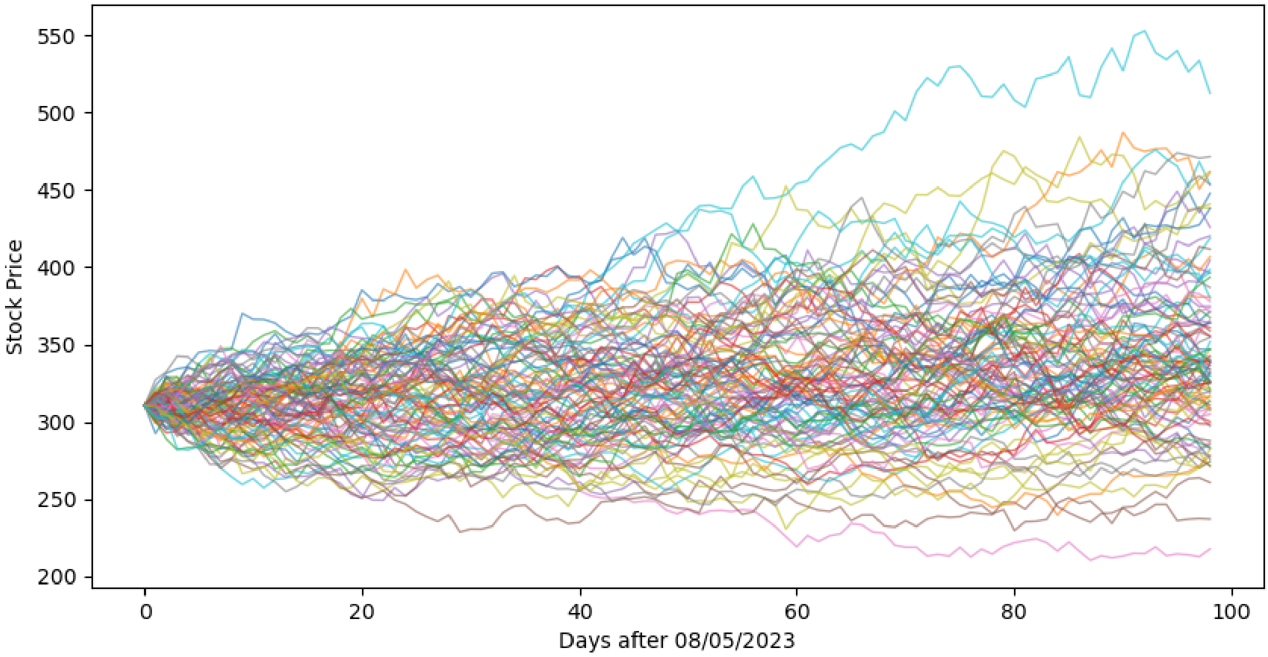


Figure 3.1: Monte Carlo Simulation of Microsoft

We will now see how the average of the 100 scenarios performed behaves, the predictions of the Linear regression model and the real variation in the stock price. Thus, the simulation carried out with the average of the Monte Carlo scenarios since 08/05/2023 indicates an increasing trend (Figure 3). On the other hand, the linear regression model predicts the stock price a little more precisely over the 100 days after 05/08/2023. But it does not take into account the random nature of stock prices, which means that it will not be suitable, for example, during periods of crisis where the variation in stock prices becomes purely random.

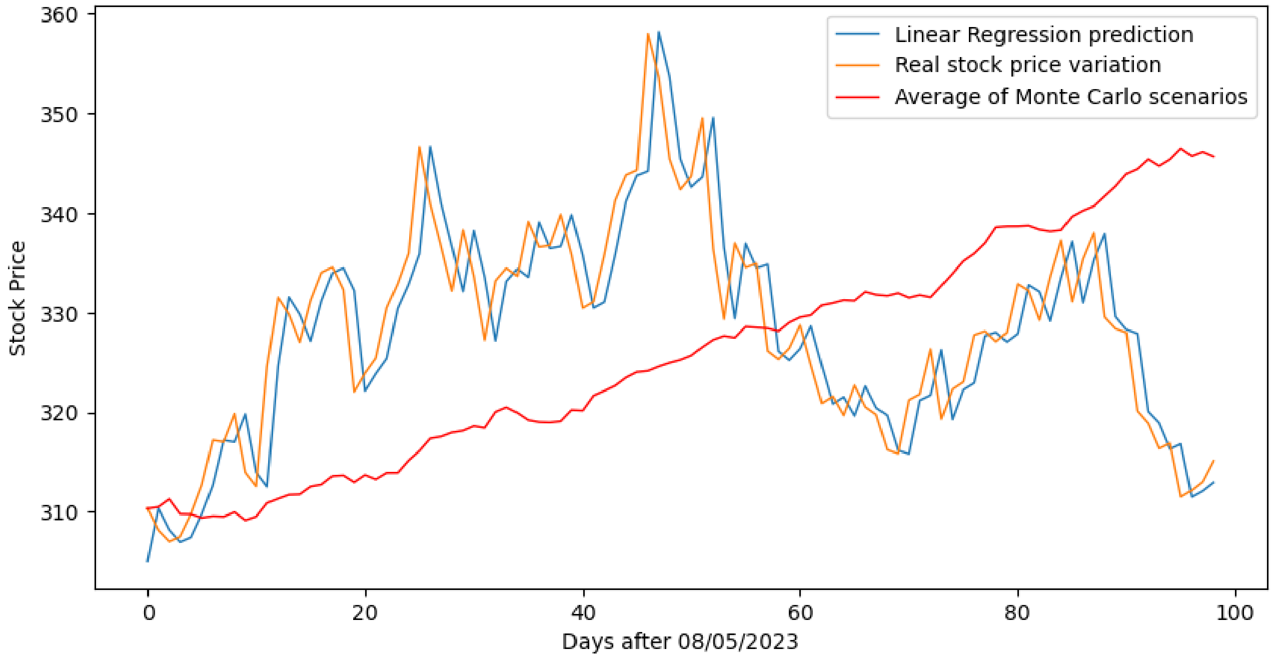


Figure 3.2: Comparison between scenarios Monte Carlo average, Linear Regression prediction and reals evolution of stock price

3.2 Source code in Python

```
pip install yfinance

import yfinance as yf
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression
from sklearn.pipeline import Pipeline
from sklearn.preprocessing import StandardScaler, PolynomialFeatures
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error

code = ’MSFT’
apple = yf.Ticker(code)
data =
   apple.history(interval=’1d’,start=’2018-01-01’,end=’2023-10-01’)
df  = pd.DataFrame(data)
header = [’Open’,’High’,’Low’,’Close’,’Volume’]
df = df[header]
df[’Next_close’] = df[’Close’].shift(-1)
```

```
df.dropna(inplace=True)
df.head()
df[’Daily yield (%)’] = (df[’Next_close’] / df[’Close’] - 1)* 100
df.head()
#dat = df[[’Open’,’High’,’Low’,’Close’,’Volume’,’Next_close’]]
#dat.corr()

x_data = df[[’Close’,’Volume’,’Daily yield (%)’]]
y_data = df[[’Next_close’]]
x_train, x_test, y_train, y_test = x_data[:1345], x_data[1346:], y_data[:1345], y_data[1346:]

#Monte carlo
#Parametres: drift, mean, variance; nombre de simulation
Data_mean = np.log(1 + x_train[’Daily yield (%)’].mean()/100)

Data_sq = (0.01*x_train[’Daily yield (%)’]).std()

drift = Data_mean - 0.5*Data_sq**2

n = 50

simulations = np.zeros((n,len(y_test)))
simulations[:,0] = y_test[’Next_close’][0]

for i in range(1, len(y_test)):
    rendements = np.random.normal(Data_mean, Data_sq, n)
    simulations[:, i] = simulations[:,i - 1] * np.exp(rendements)

temps = np.arange(len(y_test))
plt.figure(figsize=(10, 5))
#for i in range(n):
#    plt.plot(temps,simulations[i, :], label=f’Scenario {i + 1}’, alpha=0.6)

#plt.legend()
simulations_mean = np.zeros(len(y_test))
for i in range(len(y_test)):
    simulations_mean[i] = simulations[:,i].mean()

plt.plot(simulations_mean,’r’)
plt.xlabel(’Days after 08/05/2023’)
plt.ylabel(’Stock Price’)

y_data.shape

Lr = LinearRegression()
Lr.fit(x_train[[’Close’]],y_train)
Lr.score(x_train[[’Close’]],y_train[[’Next_close’]])
yhat = Lr.predict(x_test[[’Close’]])

plt.figure(figsize=(10, 5))

plt.plot(yhat,label=’Linear Regression prediction’)

plt.plot(temps,np.array(y_test['Next_close']), label='Real stock price variation')
plt.plot(simulations_mean,'r',label='Average of Monte Carlo scenarios')
plt.xlabel('Days after 08/05/2023')
plt.ylabel('Stock Price')
plt.legend()

# Error printing
print(f"Le MSE du Mod le brownien est:{mean_squared_error(simulations_mean,y_test['Next_close'])}")
print(f"Le MSE de la RL est: {mean_squared_error(yhat,y_test['Next_close'])}")
```

Conclusion

This study shows that it is possible to fill the gap the financial sector, in particular the stock market using the Monte Carlo simulation Geometric Brownian motion as underlying stock price model. As a result, the parameters of the GBM models were determined based on the market characteristics of recent years to achieve the Monte Carlo Simulation. The accuracy of such an approach was found to be meaningful using our approach. Since all objectives are obtained through the proposed system, the results can be used to help traders make a more informed decision, based on past and future trends in the stock market. Note that this method allows you to have an overview of trends and not exact prices. We therefore used linear regression to validate the trends provided by the Monte Carlo method. It turns out that the predictions made by the Monte Carlo method follow the trends but do not give more precise results like the linear regression model. But here too there is a problem with the linear regression method because it does not take into account the random aspect of prices. It would then be convenient to use a model which will perhaps combine linear regression and GBM.

BIBLIOGRAPHY

[1] Z. N. Hamdan, S. N. I. Ibrahim, and M. S. Mustafa, *Modelling malaysian gold prices using geometric brownian motion model," Adv. Math. Sci. J., 2020, doi: 10.37418/amsj.9.9.92*

[2] S. Ghahramani, "Geometric Brownian Motion, *in Fundamentals of Probability with Stochastic Processes, 3rd ed. Upper Saddle River, New Jersey, USA: Pearson, 2005, ch. 12, sec. 5, pp. 598-605*

[3] S. Dunbar, *Properties of Geometric Brownian Motion," in Stochastic Processes and Advanced Mathematical Finance, University of NebraskaLincoln*

[4] C. Pacati, *Brownian Motion and Geometric Brownian Motion, 2011*

[5] Thomas Mulc, *Modeling Stock Prices with Geometric Brownian Motion*

[6] K. Nagarajan and J. Prabhakaran, *Prediction of stock price movements using Monte Carlo simulation," Int. J. Innov. Technol. Explor. Eng.,2019, doi: 10.35940/ijitee.L2919.1081219*

[7] Fabio Lopes Lich, *Monte Carlo Method and Brownian Movement Applied to Future Stock Market Analysis*

[8] R.Y. Rubinstein and D.P. Kroese. *Simulation and the Monte Carlo Method: Third Edition. Wiley, 11 2016.*

[9] David R. Harper. *How to use monte carlo simulation with gbm, Aug 2020.*