Illinois Institute of Technology

Stuart School of Business

Master of Science in Finance

Independent Study in Finance: Spring 2025

Dynamic Volatility Forecasting for Option Pricing: Integrating GARCH(1,1) and Kalman Filters

Submitted to – **Prof. Bruce Rawlings**

Submitted By :-

1.**Vinayak Mettu**

2. **Sourabh Patil**

MS Finance, 2025

Abstract:

In this study, we propose and test a fully Bayesian state-space approach for dynamic option valuation by integrating three established elements:

GARCH(1,1): volatility forecasting. We employ a GARCH(1,1) model to generate one-step-ahead estimates of the underlying asset's conditional variance, thereby capturing key features such as volatility clustering and mean reversion.

Black–Scholes–Merton pricing. These variance forecasts are input directly into the Black–Scholes–Merton formula to produce initial theoretical prices for European calls and puts.

Kalman filtering: Treating the Black–Scholes–Merton outputs as noisy observations, a Kalman filter recursively updates the latent volatility state, its uncertainty, and the implied option prices as new market quotations become available.

Calibrating the GARCH parameters by maximum likelihood on six months of end-of-day data for 37 large-cap equities, we initialize the Kalman filter with these posterior priors. Forecast performance is evaluated by comparing filter-adjusted prices to observed mid-quotes, using both out-of-sample

R^2 and log-likelihood metrics. Our hybrid framework achieves a correlation exceeding 0.99 between predicted and actual option prices, and markedly reduces biases in the distribution tails relative to a static plug-in method.

These findings demonstrate that embedding econometric volatility forecasts within a recursive Bayesian filter creates a highly accurate and adaptive pricing engine. Nonetheless, the results also underscore the importance of incorporating market-derived volatility adjustments (e.g., smile and skew overlays) for practical trading implementations.

Introduction

During our undergraduate years, we actively traded equity and index options, which offered firsthand experience in how volatile, noisy, and difficult to model financial markets can be. This practical exposure made me deeply interested in the role of quantitative models in pricing derivatives, particularly models that can adapt in real time to reflect market sentiment and volatility. Despite the widespread use of the Black Scholes Merton (BSM) model, everyone know that its assumption of constant volatility was not sufficient to capture real-world market behavior or tail events in OTM options.

As a part of our independent study under the guidance of Prof. Bruce Rawlings, we decided to explore a Bayesian approach to option pricing. Our project was initially motivated by a prior academic paper that implemented a Bayesian approach for model option prices using synthetic data; however, its lack of real-world testing left open the question of how such a method would perform using live market data.

In parallel, our coursework in Bayesian Statistical Interference introduced us to the Kalman filter, a recursive Bayesian estimation algorithm widely applied in signal processing, control systems, and increasingly, in financial time series analysis. Kalman filter is particularly well-suited to environments with latent variables and evolving dynamics, such as financial markets-where parameters like implied volatility are unobservable and non-stationary.

This project integrates three key components:-

1. **GARCH (1,1)** to model volatility clustering
2. **Black-Scholes Merton** to price European option
3. **Kalman Filter** to dynamically estimate and update the underlying volatility and other latent variables over time.

The primary objective of this project is to develop a Bayesian dynamic system for forecasting options prices and implied volatility in real-world market conditions. We aim to assess whether Kalman-filter volatility estimates, derived from GARCH(1,1) implied priors and updated using market option prices, can produce more accurate and adaptive option price forecasts than traditional models.

To conduct this study, we collected historical stock prices through Bloomberg Terminal and option chain data using polygon.io, and we considered the risk-free rate as the 3-month treasury rate obtained using the FRED database. These allowed us to test our model on actual market behavior and evaluate predictive performance using metrics such as forecast error, variance stability, and responsiveness to market shocks.

Ultimately, this project serves as proof of concept that the Bayesian method, specifically the Kalman filter, can be effectively used in combination with traditional financial models to improve the real-time pricing and forecasting of financial derivatives

2.0 Methodology Background and Methodology

2.1 Data

Our empirical study relies on three complementary data sets, all aligned over a common six-month horizon form November 1, 2024, through March 21, 2025.

1. **Equity Prices**
   We obtained daily closing prices for 37 large- capitalization stocks via Bloomberg. These prices were used both to compute log-returns for volatility modeling and to serve as input into out option pricing routine.
2. **Risk- Free rate**
   The continuous risk-free rate was proxied by the three-month U.S treasury yield, downloaded from **FRED database.** Missing observations were forward filled to ensure complete daily that matches the equities trading calendar.
3. Option Chain Data
   End-of-day option quotes (Call and Put) for each of the 37 underlying were retrieved via Polygon.io. for every trading date in our sample, we captured strike price, time to maturity (in days), bid-ask midpoint, and traded volume.

All three series were merged on trading date. We then aligned maturities by converting each option's quoted time-to-expiry into a fraction of a year (days-to-expiry/365). This harmonized dataset forms the foundation for our GARCH(1,1) calibration, Black-Scholes plug in forecast, and subsequent Kalman filter update.

2.1 Black-Scholes Merton

BSM was introduced in 1976, revolutionizing financial markets by providing a closed-form solution for European options pricing. It assumes BSM follows geometric Brownian motion with constant drift and volatility.

The pricing formula for European call option is:

$$C = S_0 N(d_1) - Ke^{-rT} N(d_2)$$

Where,

$$d_1 = \frac{\ln(S_0/K) + (r + \frac{1}{2}\sigma^2)T}{\sigma\sqrt{T}}, \quad d_2 = d_1 - \sigma\sqrt{T}$$

The Black-Scholes formula for the price of a European put option is:

$$P = Ke^{-rT} N(-d_2) - S_0 N(-d_1)$$

Where,

$$d_1 = \frac{\ln\left(\frac{S_0}{K}\right) + \left(r + \frac{\sigma^2}{2}\right)T}{\sigma\sqrt{T}}, \quad d_2 = d_1 - \sigma\sqrt{T}$$

Here, so is the current stock price, K is the strike price, r is the risk-free rate, T is the time to maturity, σ is the volatility, and N (.) denotes the cumulative distribution function of the standard normal distribution.

Although elegant and widely adopted, the BSM model's assumption of constant volatility and log-normal asset returns fails to capture empirically observed features such as stochastic volatility, jumps, and the leverage effect phenomena that motivate the use of GARCH, local-volatility, and stochastic-volatility extensions in modern derivatives theory.

2.1.1 Implied Volatility Solver:

To back out σ_{imp} from a market price, we apply a Newton-Raphson root finder:

Define $f(\sigma) = C_{\text{BS}}(S, K, T, r, \sigma) - V_{\text{mkt}}$

where $C_{\text{BS}}(S, K, T, r, \sigma) = S\,\Phi(d_1) - Ke^{-rT}\,\Phi(d_2)$, $d_{1,2} = \dfrac{\ln(S/K) + (r \pm \frac{1}{2}\sigma^2)\,T}{\sigma\sqrt{T}}$, $d_2 = d_1 - \sigma\sqrt{T}$.

Derivative Vega: $\dfrac{\partial C_{\text{BS}}}{\partial \sigma} = S\,\varphi(d_1)\sqrt{T}$

Iteration: $\sigma_{n+1} = \sigma_n - \dfrac{f(\sigma_n)}{S\,\varphi(d_1(\sigma_n))\sqrt{T}}$

Terminate when $|f(\sigma_n)| < 10^{-8}$ or after 200 steps.

Validation and Speed test: We include small helper routines to (a) verify that bs_call & find_vol reproduce known volatilities on synthetic test cases, and (b) benchmark performance over 10,000 random inputs.

Appendix B contains the complete Python source for black-Scholes, bs_call, bs_vega, and find_vol.

2.2 GARCH [1,1]

The Generalized Autoregressive Conditional Heteroskedasticity (GARCH) model, developed by Tim Bollerslev proposed GARCH model that incorporates past variances into volatility dynamics. GARCH (1,1) model – a model with one lag of both variance & error term has become a workhorse for volatility modeling and forecasting due to simplicity and effectiveness. It captures a key observation that volatility is conditional on past information and changes over time rather than remaining constant.

The GARCH (1,1) formula:

$$\sigma_t^2 = \omega + \alpha\epsilon_{t-1}^2 + \beta\sigma_{t-1}^2$$

Where,

Omega (ω) :- Constant term, often interpreted as baseline or long run variance level.

Alpha (α) :- Coefficient of ARCH term ϵ_{t-1}^2 Higher alpha means that volatility reacts more strongly to

yesterday's return shock.

Beta (β) :- Coefficient of GARCH term σ_{t-1}^2 Higher β implies that volatility is more persistent over time.

Take expectation on both side:

$$\mathbb{E}[\sigma_t^2] = \omega + \alpha\mathbb{E}[\epsilon_{t-1}^2] + \beta\mathbb{E}[\sigma_{t-1}^2]$$

Assuming $\epsilon_{t-1} \sim \mathcal{N}(0,\sigma_{t-1}^2)$, **then** $\mathbb{E}[\epsilon_{t-1}^2] = \mathbb{E}[\sigma_{t-1}^2]$ **so:**

$$\mathbb{E}[\epsilon_{t-1}^2] = \mathbb{E}[\sigma_{t-1}^2]$$

Let $\bar{\sigma}^2 = \mathbb{E}[\sigma_t^2]$**, then:**

$$\bar{\sigma}^2 = \omega + (\alpha + \beta)\bar{\sigma}^2 \Rightarrow \bar{\sigma}^2(1 - \alpha - \beta) = \omega \Rightarrow \bar{\sigma}^2 = \frac{\omega}{1 - \alpha - \beta}$$

All parameters must be non-negative $\omega \geq 0, \alpha \geq 0, \beta \geq 0$ and the key condition for stationarity is $\alpha + \beta < 1$

2.2.1 Capturing Volatility Dynamics

1. By including ϵ_{t-1}^2, GARCH (1,1) captures news impact large return increases r^2t-1n, thus boosting" σ_{t-1}^2 , which means volatility increases after turbulence.
2. σ_{t-1}^2 carries forward past volatility clusters
   - Burst volatility followed by higher volatility
   - Calm produces low volatility, unless disturbed by a new shock.
3. Overall, if no shock occurs, the old shock decays geometrically as volatility reverts to its mean [long-run level].

Asset	omega	alpha	beta
SPX	0.000002	0.100000	0.880000
AAPL	0.000005	0.045000	0.935390
ABNB	0.000084	0.167298	0.743317
ADBE	0.000236	0.019465	0.569176
AMAT	0.000014	0.011407	0.968404
AMD	0.000021	0.010000	0.970000
AMZN	0.000057	0.280565	0.656321
T	0.000115	0.104248	0.444341
AVGA	0.000015	0.050000	0.930000
BA	0.000014	0.050688	0.925511
BLK	0.000006	0.050000	0.930000
CAT	0.000140	0.066729	0.523662
COF	0.000060	0.185021	0.713279
COST	0.000004	0.045207	0.937807
CRM	0.000011	0.010000	0.970000
CVX	0.000005	0.050000	0.930000
DASH	0.000025	0.100000	0.880000
DIS	0.000042	0.070928	0.819288
FDS	0.000004	0.010000	0.970000
FDX	0.000137	0.056532	0.662436
GOOG	0.000008	0.009996	0.969655
GS	0.000154	0.315315	0.204170
IBM	0.000198	0.027233	0.000000

JNJ	0.000095	0.199473	0.000000
JPM	0.000005	0.050000	0.930000
MA	0.000004	0.100000	0.880000
META	0.000018	0.010000	0.970000
MS	0.000136	0.194260	0.415559
MSFT	0.000006	0.045269	0.935479
NEE	0.000032	0.050000	0.850000
NFLX	0.000019	0.010000	0.970000
NVDA	0.000070	0.025397	0.916582
ORCL	0.000116	0.090000	0.639997
PLTR	0.001131	0.095349	0.303428
RL	0.000035	0.073884	0.867385
TGT	0.000161	0.010764	0.720072
TSLA	0.000030	0.010000	0.970000
UAL	0.000055	0.091404	0.850564
UBER	0.000006	0.011347	0.980817
WMT	0.000004	0.010000	0.970000
XOM	0.000006	0.050000	0.930000

Table 1. GARCH (1,1) Results

2.3 Kalman Filters

The Kalman Filter, introduced by Rudolf Kalman in 1960, is a recursive algorithm designed to estimate the hidden state of a linear dynamic system based on a series of incomplete and noisy measurements. Kalman filters require an estimate of the State variance for the observation state. We set discount factor d1 in Kalman filter between 0.8 to 1. A lot of notions are required for Kalman recursion. Estimates of the covariance of disturbance in the system equation are required.

It consists of two stages:-

I. Prediction Stage: Forecast the next step and its covariance based on system dynamics
II. Update state: Correct the forecast using the new observation and Bayesian updating principles.

Notations:-

The number of parameters is n.

The number of variables to predict at t is r.

Observation Model:

$$y_t = F_t^\top \theta_t + \nu_t, \quad \nu_t \sim \mathcal{N}(0, V_t)$$

Y_t = [r x 1] observable scalar at each time.

$F_t^{'}$ = [n × 1] design matrix of independent variable

θ_t = [n × 1] system or state vector

- In a regression setting, these are the betas of the equation.

ν_t = In a regression setting, these are the regression errors.

State System Equation:

$$\theta_t = G_t \theta_{t-1} + \omega_t, \quad \omega_t \sim \mathcal{N}(0, W_t)$$

G_t = [n × n] system matrix

ω_t = system error or disturbance, which has a covariance of ω_t

G = evolution or state matrix

- In finance, these are identity matrix
- If G is the Identity matrix
- The system equation referred to a "**random walk"** of the coefficients.

Prediction step:

$$a_t = G_t m_{t-1}, \quad R_t = G_t C_{t-1} G_t^\top + W_t$$

m_{t-1} = a priori estimate

a_t = the exploration of m_{t-1} for a new estimate of θ_t

G_t = the known state transition matrix for θ

Observation Forecast:

$$f_t = F_t^\top a_t, \quad Q_t = F_t^\top R_t F_t + V_t$$

Kalman Gain:

$$A_t = \frac{R_t F_t}{Q_t}$$

Update step:

The optimal amount of error is used to update:

$$m_t = a_t + A_t(y_t - f_t)$$

Update the covariance of

$$C_t = R_t - A_t A_t^\top Q_t$$

Variables used: Multivariate DLM / Kalman filters

Variable	Symbol	Dimensions
Prior Volatility	a_t	(n × 1) state vector
Posterior volatility	m0	(n × 1) state vector
Prior variance	R_t	(n × n) state covariance
Posterior Variance	C0	(n × n) state covariance
Forecasted Option price	f_t	(1 x 1) Scaler
Observation error	e_t	(1 x 1) Scaler
Observation Variance	Q_t	(1 x 1) Scaler
Degree of freedom	n_{t-1}	(1 x 1) Scaler
Sum of squared errors	d_{t-1}	(1 x 1) Scaler
State-noise variance	W_t	(n x n) system covariance
Variance Discounting Factor	δ (d_1)	(1 x 1) Scaler
Omega	ω	(1 x 1) Scaler
Alpha	α	(1 x 1) Scaler
Beta	β	(1 x 1) Scaler
Identity Matrix	Wscale	(1 x 1) Scaler
System (evolution) matrix	G_t	(n × n), often I_n
Design vector	F_t	(n × 1) design/regressor
Kalman gain	A_t	(n × 1) error-amplification
Discount vector	δ_2	(n × 1) scalar or vector
Discount diagonal matrix	D_t	(n × n), diag(δ_2)

Table 2.0 Variables used in multivariate Kalman Filter

2.5 Synthetic Data Generation

Objective. To assess how well our Kalman–GARCH–BSM framework recovers known dynamics, we generate synthetic option chains alongside real-world data.

Procedure:-

Return Simulation: We simulate daily log-returns via an AR(1) process with GARCH(1,1) innovations, using parameters $\{\omega,\alpha,\beta\}$ estimated from historical equity returns.

Price Path Construction: Returns are cumulated to form a price series St. We retain the final N=60 days to match our real-data horizon.

Strike Selection: The last N N prices are sorted; we then sample K equally spaced quantiles and round each to the nearest integer (e.g. 129.86 → 130). This produces a realistic cross-section of in-, at-, and out-of-the-money strikes.

Implementation details and full code appear in **Appendix A**.

3.0 Kalman Filter Framework

Here we initialize the Kalman filter for option data

We create data matrices for the Kalman filter:

I. Prior mean (_), posterior variance (_)
II. Posterior mean (_), posterior variance (_)
III. Forecasted price (_), Forecasted error (_)
IV. Initialize noise variance (Wscale)
V. We set up our priors and posteriors

Why are we doing :

1. Predict Step
   - Forecast Next day volatility using GARCH (1,1)
   - Update the uncertainty about the volatility (R)

2. Forecast step
   - Use the Black-Scholes formula with predicted volatility to forecast Option Prices
   -
3. Update step
   - Compare the forecasted option price to the observed option price
   - Compute the forecast error (e)
   - Update posterior estimate (m0 and C0)
   - Update the Variance of error (Q)

4. Log Likelihood
   - Keep track of how likely the observed prices are given the model.

5. Plot and analyze the result
   - Volatility by Day
   - Implied vs Prior GARCH(1,1) Volatility
   - Observed Option and Forecast Price by Day
   - Observed Option and Forecast Option Price

3.1 What exactly going under RunDLM:-

Here we perform the **Predict step** to forecast the next day's volatility using GARCH(1,1) model, which captures the time-varying nature of volatility based on past returns. This predicted volatility is essential for future option prices. Along with forecasting volatility, we also update the uncertain associated with this volatility prediction, denoted as **RRR,** which represents the variance of the prior estimates and accounts for the randomness introduced through GARCH process and process noise.

In the **Forecast step,** we use predicted volatility from previous step and input it into the **Black-Scholes formula** to forecast the option prices. Since option prices is highly sensitive to volatility, accurately forecasting volatility enables us to generate realistic prediction of the next day's option market price.

The **Update step** is crucial for refining our model estimates. Here, we compare the forecasted option price with the actual observed option price from the market. The difference between them is **called forecasted error (e).** using this error, we update the posterior estimates of the state variables, like mean mmm(updated volatility) and variance c (updated confidence in the volatility). Additionally, we update the variance of the error forecast (Q) to reflect new information about the observation noise. This step ensures that our current model continuously learns and corrects itself as new data becomes available.

We also calculate **Log Likelihood** at each time step. The log likelihood measures how probable the observed market prices are, given the model's current state and forecast. By accumulating the log likelihood over time, we can evaluate the overall fit of the model, and later, we can use this likelihood as a criterion to optimize model parameters if needed.

Finally, after running the model, we **Plot and analyze** the result to visually assess model performance. We Generate plots such as Volatility by day (comparing implied, prior GARCH, and posterior Kalman volatilities), Implied vs Prior GARCH(1,1) Volatility scatterplots to check relationship, Observed Option and forecasted option price by day line plots, observed vs Forecasted Option Prices scatterplots to evaluate prediction quality. These visual diagnostics provide intuitive and critical insights into the behavior and accuracy of the forecasting system.

3.2 Parameter Calibration

OLS method is used to update the prior mean and posterior variances.

We used m0=0.16 and C0=0.1, but it is working for a few stocks which are liquid and ATM For OTM, it is not working the way we want it to work. We will now provide some graphs.

So, what we did to update our model. We used the Ordinary Least Squares method (OLS) by sacrificing the first few observations

- Regression :

$$y = X\beta + \varepsilon$$

- Compute Ordinary Least Squares to estimate Betas (β)

$$\hat{\beta} = \left(X'X\right)^{-1} X'y$$

- Then we compute the error from the regression

$$e = y - X\hat{\beta}$$

- Compute Variance

$$s^2 = \mathrm{var}\left(e\right)$$

- Set up the initial value at time 0 and

$$m_0 = \hat{\beta} \quad \text{and} \quad C_0 = s^2\left(X'X\right)^{-1}$$

- If the coefficients are changing, the value of C0 provided by ordinary least squares is **too optimistic (too small).** Try inflating it by a factor of 2 or 3.

Plots & Diagnostics

Figure 1. **Daily Volatility Trajectories: Implied vs Prior (GARCH) vs Posterior (Kalman)**

This time-series plot overlays market-implied volatility (blue), the one-step-ahead GARCH(1,1) forecast (green), and the Kalman-filtered posterior (red), showing how the filter smooths and corrects the raw GARCH estimate over time.

Figure 2. **Implied vs Prior GARCH(1,1) Volatility**

A scatter of implied vs prior GARCH forecasts, with a fitted regression line; deviations from the 45° line quantify bias before applying the Kalman update.

Figure 3. **Observed Option and Forecast Price by Day**

A daily overlay of actual mid-quote option prices (dots) and the Kalman-filtered forecasts (solid line), demonstrating the model's tracking performance over the sample period.

Figure 4. **Observed Option vs Forecast Option Price**

A one-to-one scatter of observed vs forecasted prices; the tight clustering around the 45° line indicates high predictive accuracy of the dynamic model.

1. APPLE

AAPL_210_call

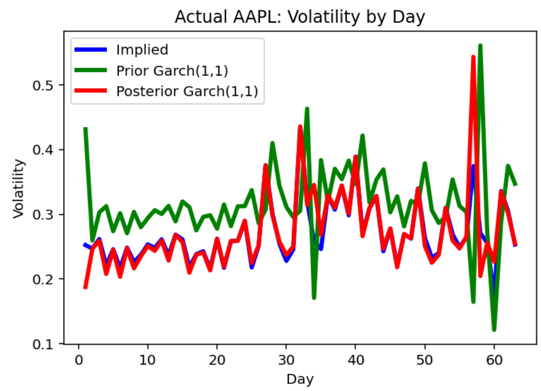

Fig. 1.1.1

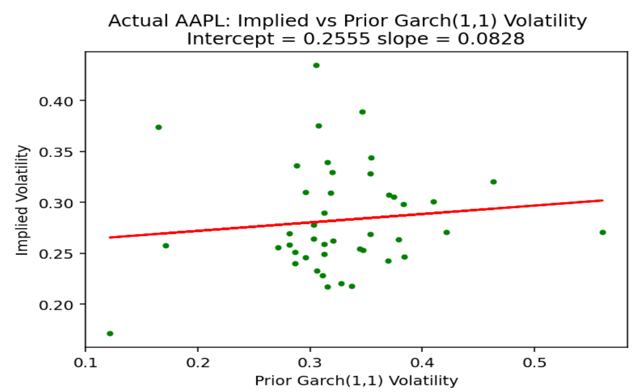

Fig 1.1.2

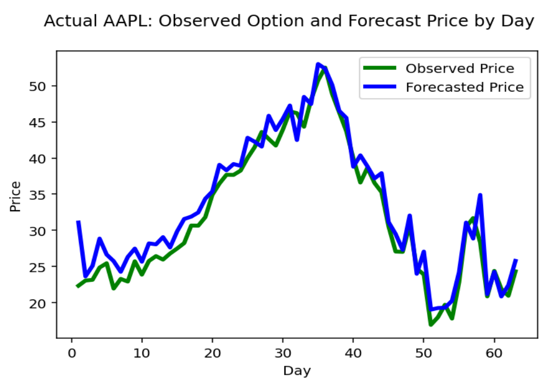

Fig 1.1.3

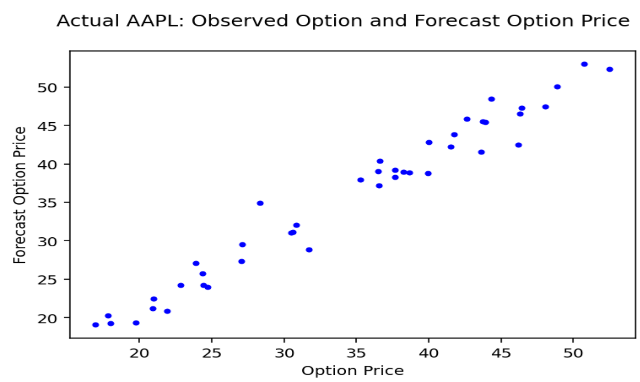

Fig 1.1.4

AAPL_220_call

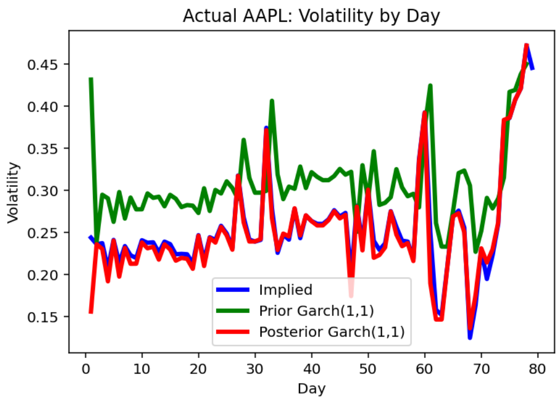

Fig 1.2.1

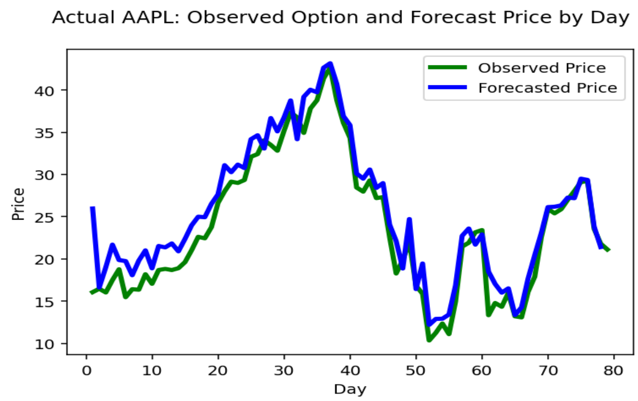

Fig 1.2.2

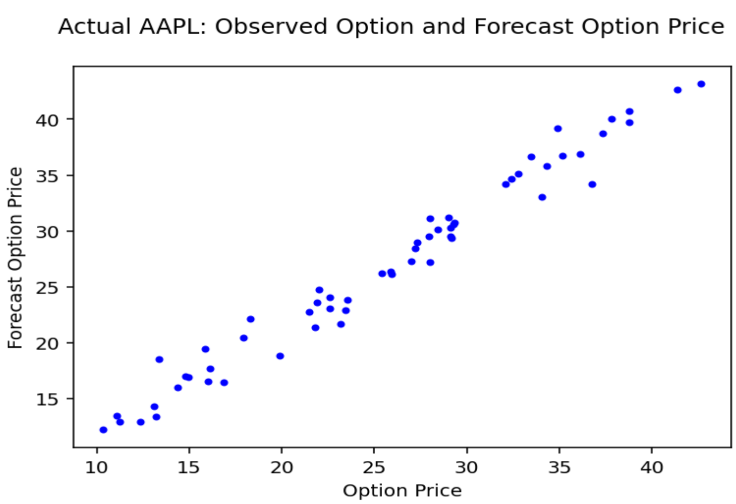

Fig 1.2.3

AAPL_230_call

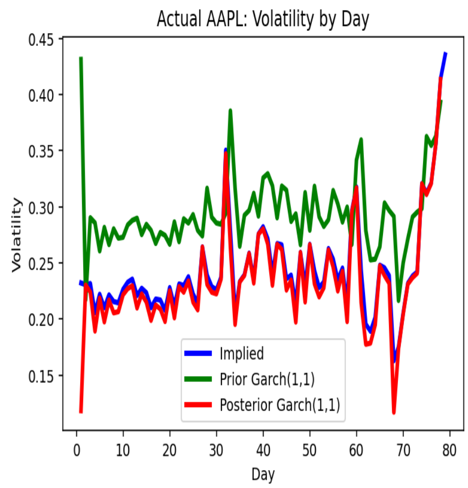

Fig 1.3.1

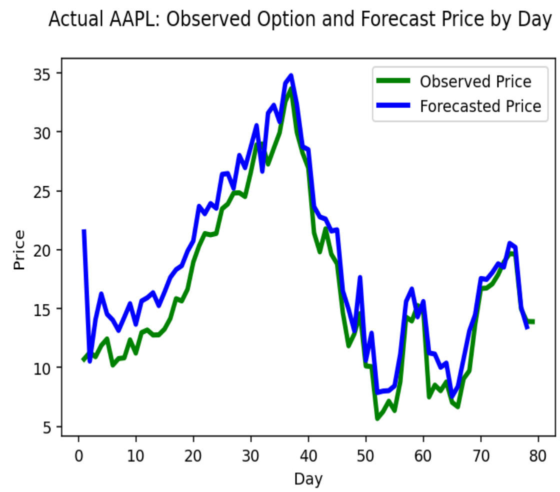

Fig 1.3.2

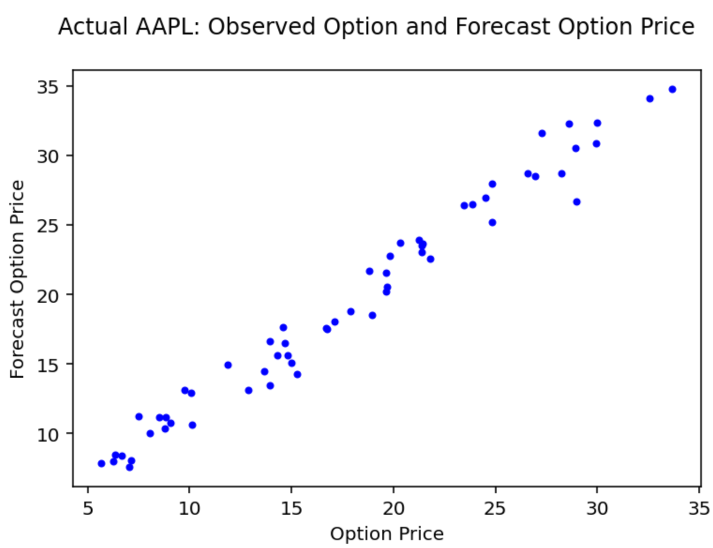

Fig 1.3.3

2. MSFT

MSFT_420_call

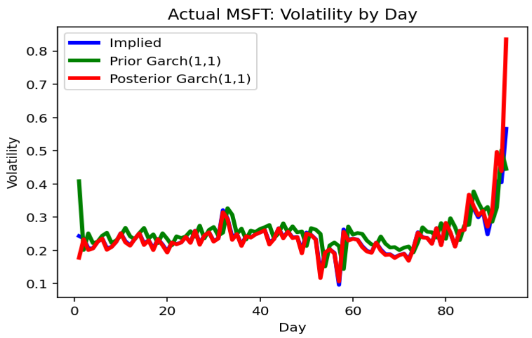

Fig 2.2.1

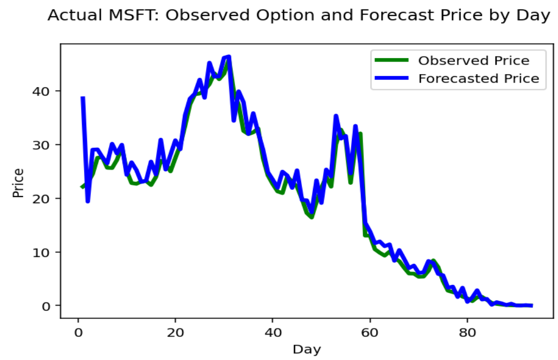

Fig 2.1.2

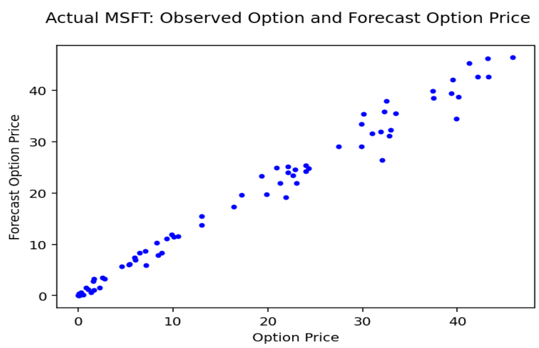

Fig 2.1.3

MSFT_440_call

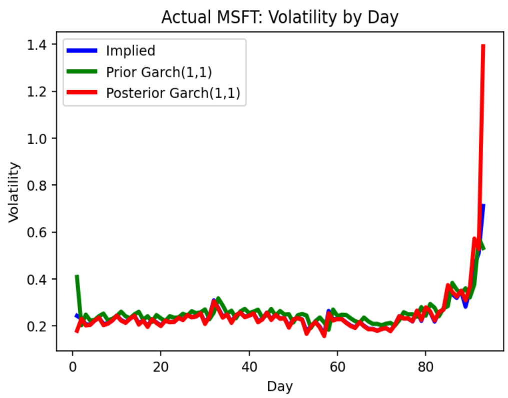

Fig 2.2.1

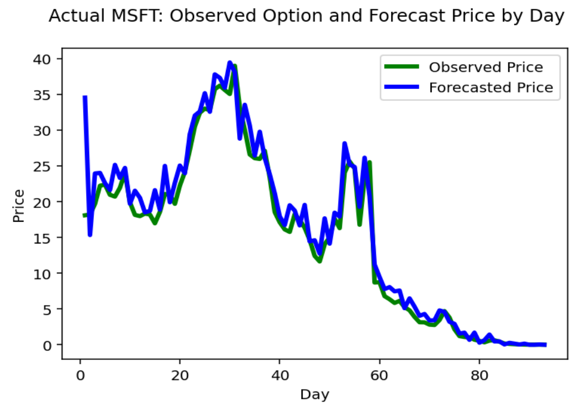

Fig 2.2.2

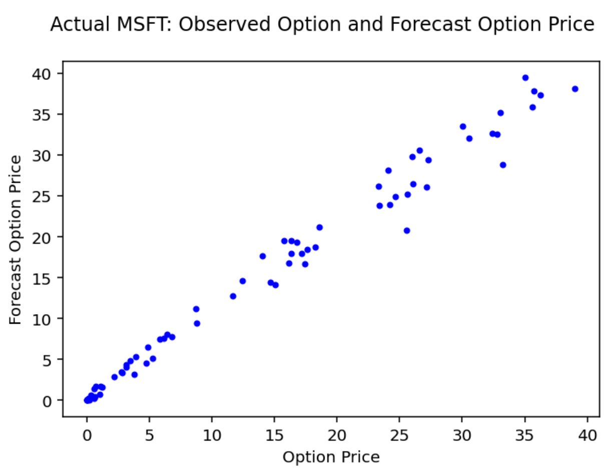

Fig 2.2.3

MSFT_500_call

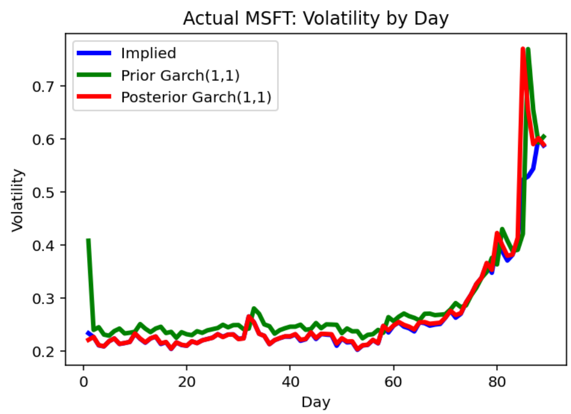

Fig 2.2.1

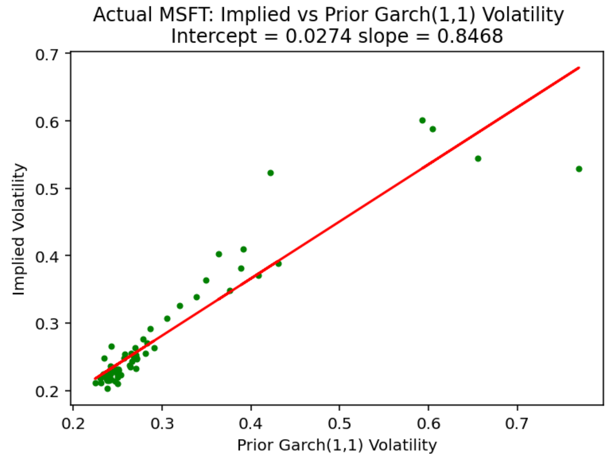

Fig 2.2.2

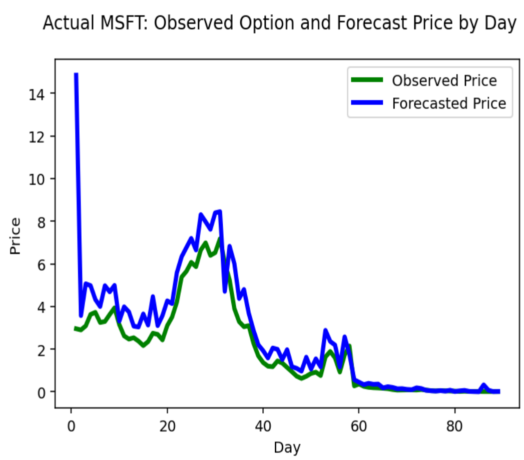

Fig 2.2.3

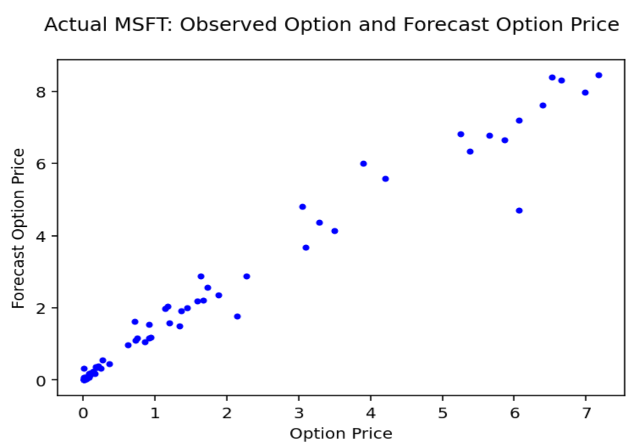

Fig 2.2.4

MSFT_460_call

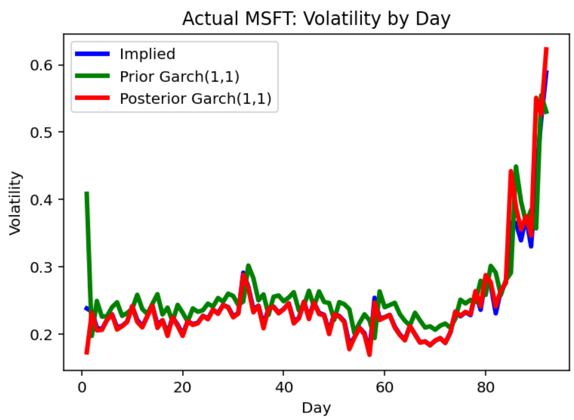

Fig 2.3.1

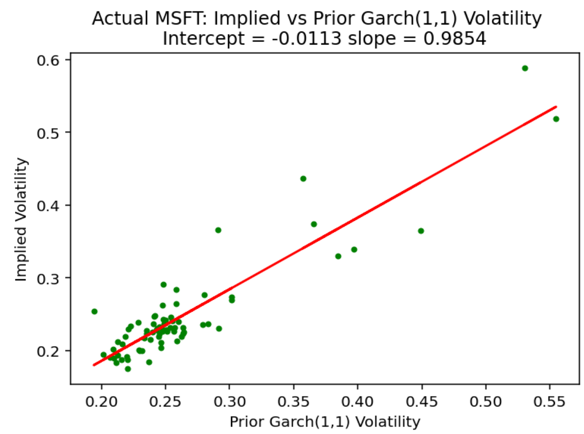

Fig 2.3.2

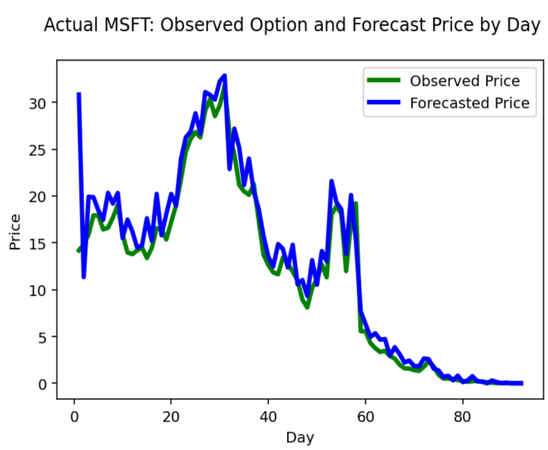

Fig 2.3.3

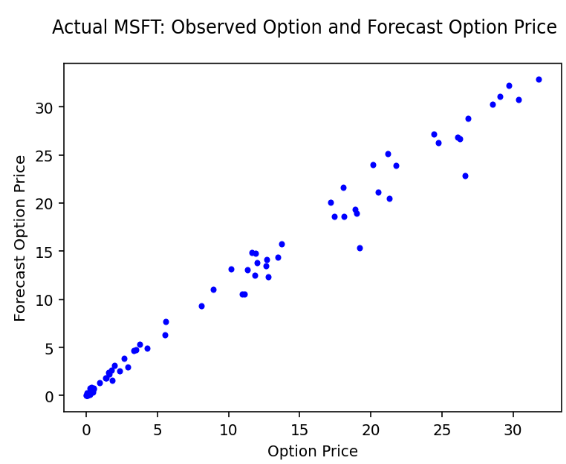

Fig 2.3.4

3. TSLA

TSLA_240_call

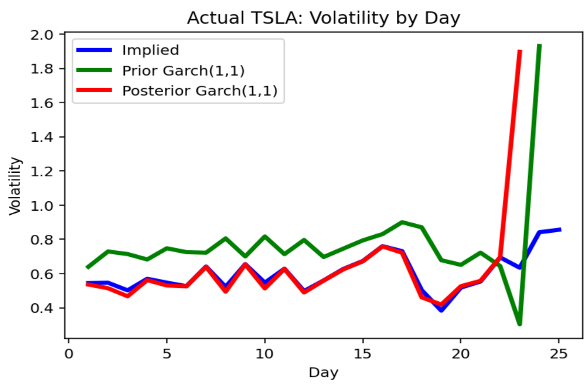

Fig 3.1.1

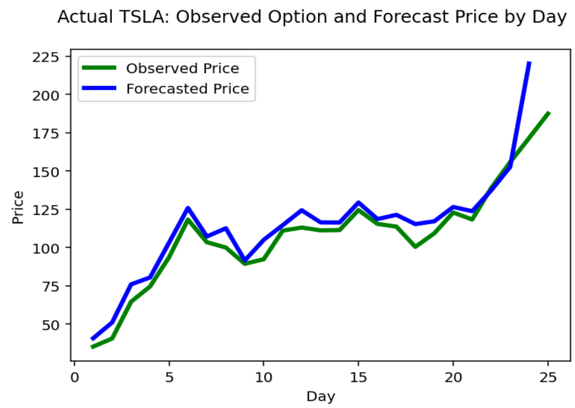

Fig 3.1.2

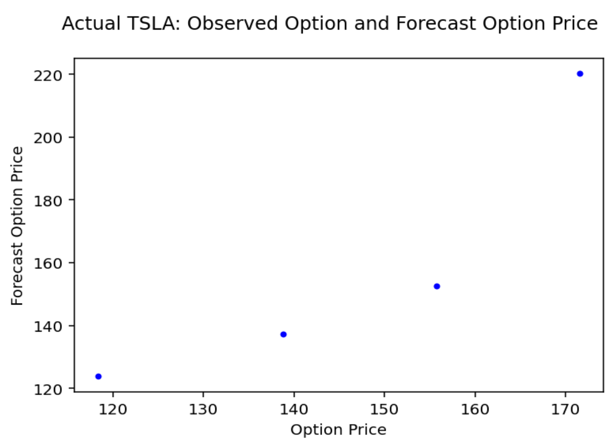

Fig 3.1.3

TSLA_380_call

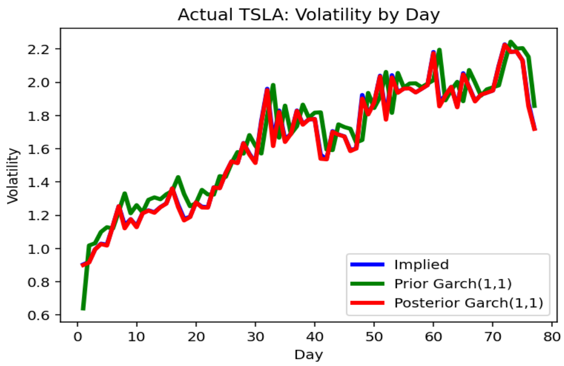

Fig 3.2.1

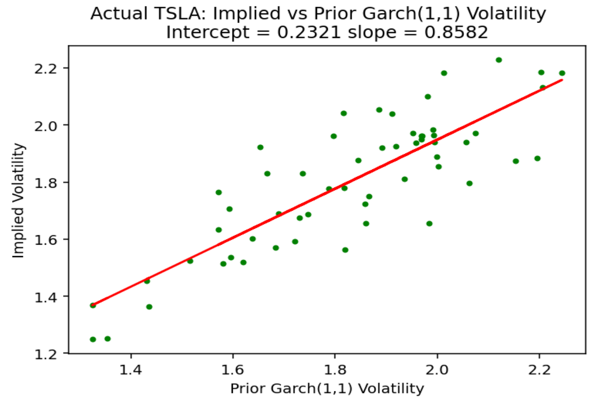

Fig 3.2.2

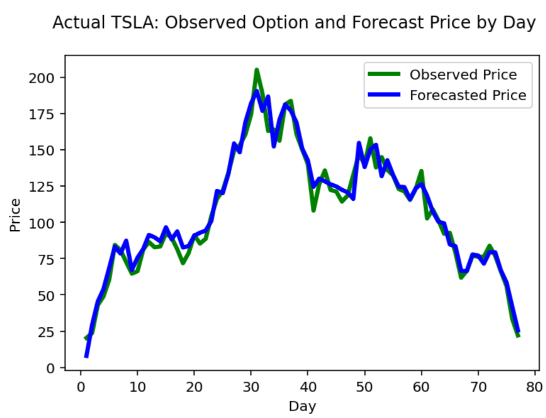

Fig 3.2.3

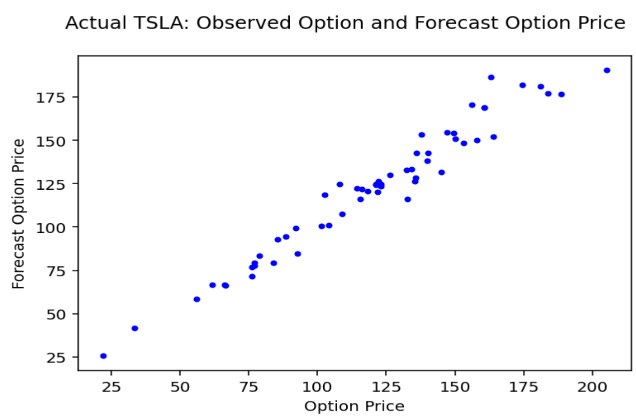

Fig 3.2.4

TSLA_320_call

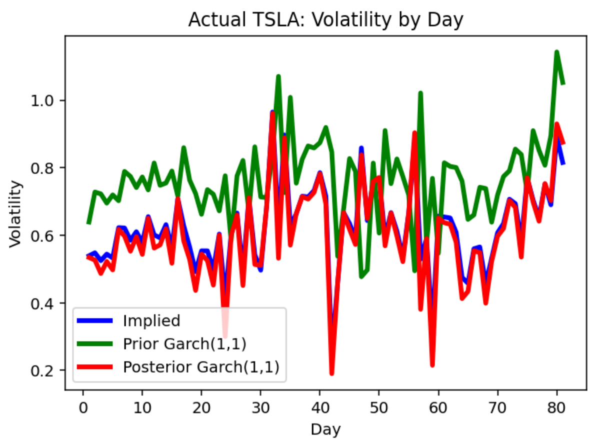

Fig 3.3.1

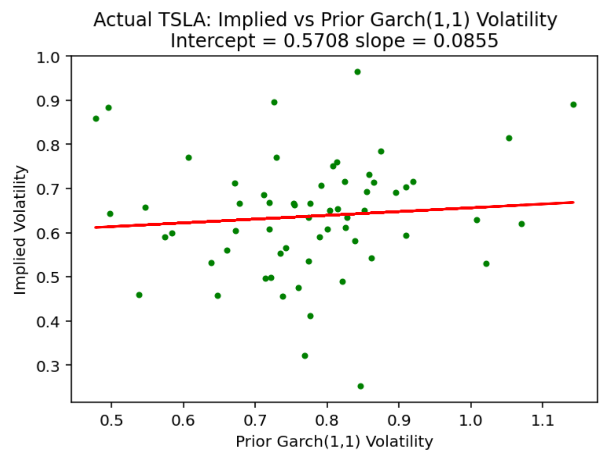

Fig 3.3.2

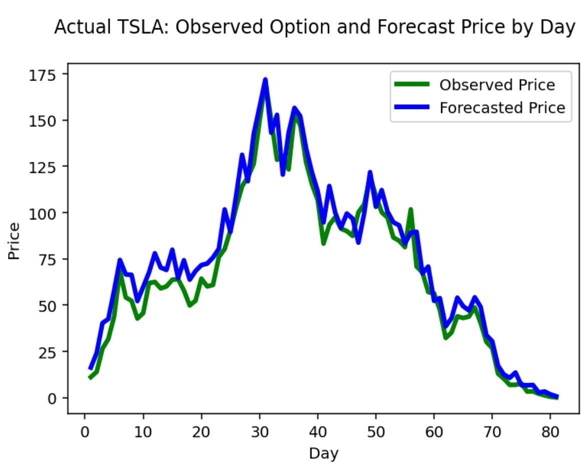

Fig 3.3.1

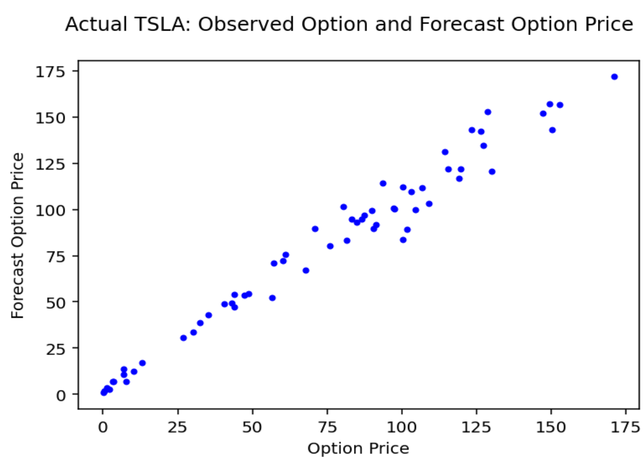

Fig 3.3.2

TSLA_360_call

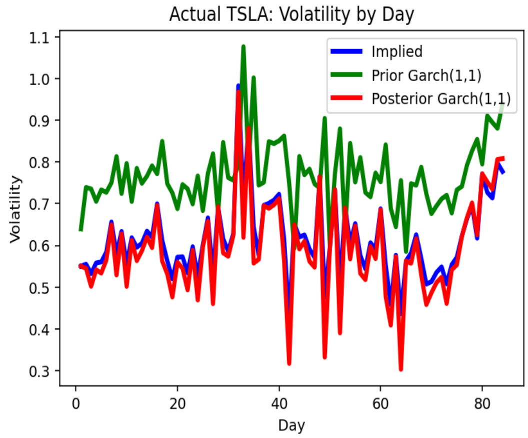

Fig 3.4.1

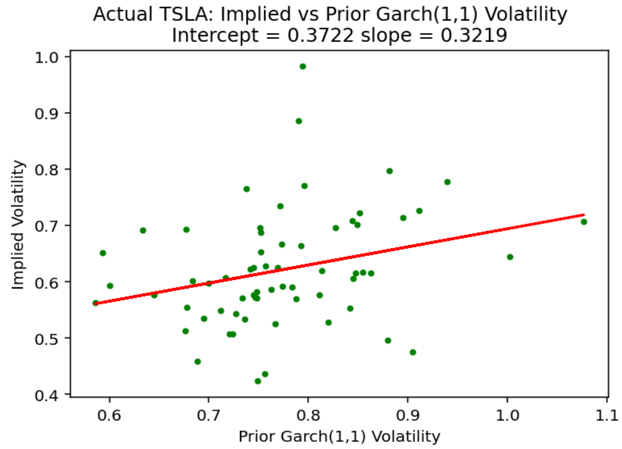

Fig 3.4.2

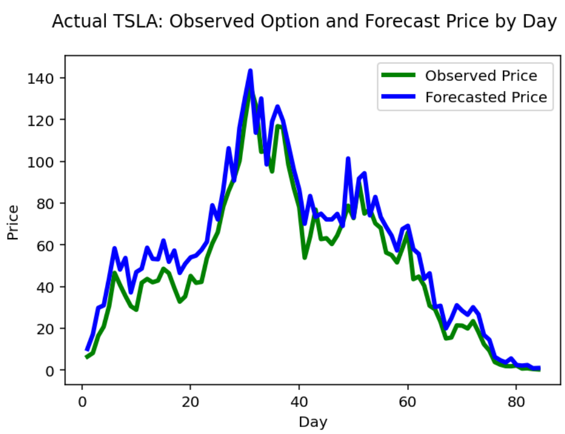

Fig 3.4.1

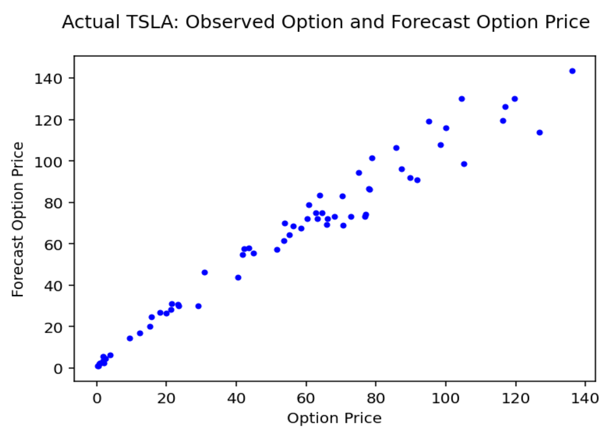

Fig 3.4.2

Result Analysis :-

The GARCH(1,1) model was used to estimate for all stocks in our list and the output provided the parameters for **omega(ω), alpha (α), and beta(β)**. The omega parameters, which captures the long-term average variance, was found very small across all assets, typically ranging between **0.000002 and 0.0002**. This behavior is expected because daily asset return generally exhibit very low conditional volatility levels.

The alpha parameters, which measures the sensitivity of volatility to new stocks, showed considerable variation across all assets. Certain Stocks such as **GS** (Goldman Sachs) and **COF** (Capital One) exhibited very high alpha measures (**>0.18**), their volatility is highly reactive to immediate market shocks. On the other hand , many technologies related stocks like **CRM** (Salesforce), **GOOG** (Google), **META** (Meta Platform), and **NFLX** (Netflix) showed very low alpha (around **0.01**), implying that their volatility are relatively stable and less sensitive to day to day fluctuations.

In term of beta parameter, which captures the persistent of volatility over time, many stocks such as **AMAT** (Applied Materials), **AMD** (Advanced Micro Devices), **CRM** (Salesforce),**META** (Meta Platforms), and **NFLX** (Netflix) have very high beta (around 0.93 to 0.98). This shows volatility for this asset is very persistent and take long time to decay. And for assets like **GS** (Goldman Sachs), **PLTR** (Palantir Technologies), IBM, and **JNJ** (Johnson & Johnson), the beta values were significantly lower, and for some (**IBM** and **JNJ**), beta was estimated at zero, suggesting that volatility shocks are absorbed quickly and that these assets revert to their mean volatility much faster.

But **PLTR**(Palantir Technologies), which has high omega (0.001131) compared to other assets, which implies significant high volatility compared to other assets. Which implies PLTR's nature of high-risk, high growth company.

Overall GARCH(1,1) estimates that different sectors have distinct volatility behaviors. Stocks generally show high sensitivity to immediate market shocks (higher alpha), technology stocks demonstrate long term volatility persistent (higher beta), and consumer staple or energy stocks tend to show moderate behaviors. These findings align well with economic intuition about behavior of different asset classes under changing market conditions.

Result Analysis of SetupDLM observation:-

In the SetupDLM stage, we start by storing the estimates GARCH(1,1) parameters as D['Garch11Param']= [omega, alpha, beta] = **[1.35775317, 0.19418434, 0.41532698].** These values represent the baseline volatility, how sensitive the model is to new market shocks, and how persistent those shocks are over time. The risk free rates over the 60-day option horizon, stored in D['Risk-free'], are quite stable , around 4.2% annually with only minor daily fluctuations. We also set the time to maturity for the options at 60 days.

The observed option prices, stored in D['y'], form a vector of 60 values, it starts around 23.5 and declines to about 6.5, which typical as options loses value closer as they get to expiration due to time decay(non-linearity). The hidden state we are trying to model, defined as D['r'] = 1, is one dimensional focusing on the underlying volatility.

At the start, the prior mean estimates D['a'] and prior variances D['R'] are set to NaN, indicating they will be filled during the Kalman filtering process. The posterior mean D['m'] are all initialized to 1, a neutral assumption, while the posterior variances D['C'] are set to zero, reflecting high confidence initially that will be adjusted as mew information comes in.

For the noise structure, the state noise variance factor D['Wscale'] is set to 0.1, and D['W'] is [1x1] matrix with value of 0.1. other variable such as the forecasted option prices D['f'], forecasted error variances D['Q'], forecasted errors D['e'], and log likelihood D['LLT'] are initialized as NaNs, ready to be updated during each filtering step.

Finally, the model priors for volatility are set. The prior mean D['m0'] = 0.001206 based on the sample variance of stock returns, and the prior variances D['C'] = 0.000402, about one-third of the prior mean reasonable level of initial uncertainty. With all these setting in place, the Kalman filter is ready to dynamically learn and adapt as it processes option price data over the 60-day horizon.

Result Analysis of RunDLM observation:-

In the RunDLM phase, we kicked off the Kalman filter's updating process the initial GARCH(1,1) parameters. The model starts with a prior mean D['m0'] and d['C0'] is 0.16 and 0.1. Even though LLT is NaN, local log-likelihood were calculated at each time step and gave valuable insights.

Our model actually did very fantastic job tracking market prices, with very high correlation of 0.9948 between forecasted and observed option prices. We also saw estimated observation variance D ['Sv'] evolve over time, starting small 2.00 x 10^-2 and growing larger as the options got closer to expiration, shows greater uncertainty.

We ran regression analysis comparing the model's forecasted prices to the actual market price. The regression coefficient D ['bhat'] intercept of around 1.13 and slope near 1.0, meaning the forecasted were very close to being unbiased and accurate. With t-statistic around 73.6 for the slope of R^2 value of 0.9896, it's clear that model explained almost all the variability in the market prices. Residuals were small and centered around zero, and the Durbin-Watson statistic dw = 1.23 indicating mild positive autocorrelation. The standard error sige of 0.716 confirmed that the prediction error was minor.

Log-likelihood term D ['LLT'], they started off moderately negative around(-14 to -19) but grew more negative as the options neared expirations, which makes sense because uncertainty increases as time runs out.

Overall, the RunDLM results show that by combining Kalman filtering, GARCH volatility modeling, and Black-Scholes pricing, we built a powerful, accurate system that tracks option price dynamics extremely well over time.

Discussion:

Our plug-in approach uses one-step-ahead volatility forecasts from a GARCH(1,1) model denoted by m_{t-1} within the Black–Scholes–Merton (BSM) pricing formula, updated recursively via a Kalman filter framework. By simulating data alongside real asset returns, we verify that the Kalman filter quickly corrects for mis-specified priors and smooths the estimated variance trajectory. This dynamic linear model perspective elegantly captures volatility clustering and mean reversion, offering a theoretically sound and computationally efficient route to option valuation.

However, when we translate this methodology into the trading pit, a new layer of complexity emerges: model risk. Real-world option market makers cannot simply plug in model-implied volatility—if they did, persistent under- or over-pricing (relative to market convention) can translate into career-ending P&L losses. Since the strike, risk-free rate, and time to expiration are exogenous inputs, the only free lever left is implied volatility itself. In practice, dealers "stretch" the volatility surface—manifesting as the familiar volatility smile or skew—to reflect supply/demand imbalances, risk premium considerations, and hedging costs. This voluntary adjustment ensures that the model's theoretical arrows align with the market's quiver: it mitigates adverse selection and steers the book toward neutrality rather than relying wholly on a statistical forecast.

Thus, while our GARCH–Kalman–BSM hybrid rigorously models the latent variance process, it also highlights an important behavioral insight: **volatility is not just a statistical artifact but a strategic variable.** A robust options-pricing engine must combine econometric rigor (to capture conditional heteroskedasticity) with a market-calibrated overlay (to guard against mispricing and protect the trader's P&L). Future extensions might incorporate implied-volatility feedback loops—feeding observed market smiles back into the state-space model—or Bayesian updating schemes that explicitly penalize divergence from prevailing implied-volatility surfaces.

Conclusion:-

In conclusion, this project highlights a fundamental challenge in financial modeling: the real world often doesn't behave the way our models assume it does. Our framework—combining GARCH(1,1), Black-Scholes, and Kalman Filters—relies on the idea that volatility follows a stochastic process (as GARCH suggests), and that this volatility is reflected in option prices through the Black-Scholes framework.

However, in the actual market, volatility is not just a function of past returns; it's also manually adjusted by market participants, particularly market makers. The Black-Scholes model, in its original form, assumes constant volatility, independent of strike price. But in practice, traders quickly learned that options far in-the-money or out-of-the-money carry tail risks that constant-volatility models do not capture. This gave rise to the volatility smile, an empirical adjustment that reflects market-implied beliefs rather than model-driven forecasts.

Because our model does not account for this structural market behavior, it struggles to match real-world option prices especially in the tails. Some of the early discrepancies in our plots also stem from the influence of poorly specified priors. One of the strengths of the Kalman Filter, however, is its ability to rapidly correct those priors as new data comes in. It "wants to work" and given a fair set of assumptions, it does. But the core problem is that the world doesn't behave according to the assumptions embedded in this model.

So while this model is mathematically sound and elegant, its failure to fully capture the adaptive, human-driven nature of financial markets limits its real-world applicability. Still, the exercise has been invaluable in showing where theory meets (and diverges from) practice and that's where the most interesting questions lie.

Bibliography:

1. Proof of the Black - Scholes pricing formula for European Call Option, https://math.stackexchange.com/questions/900911/proof-of-the-black-scholes-pricing-formula-for-european-call-option.
2. Fitting GARCH (1,1) Model, https://stats.stackexchange.com/questions/367722/fitting-a-garch1-1-model
3. Bayesian Econometrics, Gary Koop