MTH517: Time Series Analysis

Course Project

Group No. 11
Aman Agrawal 190102
Moin Ahmed 190510
Raghib Anwar 190661
Rohit Kejriwal 190717

Indian Institute of Technology, Kanpur

Option Pricing using Time Series Analysis

November 18, 2021

Acknowledgement

We would like to express our gratitude to Professor Dr.Amit Mitra to show us various ways to complete the project and have a sound theoretical background, having shared his valuable lecture notes moreover by clearing all doubts putforth by us. We would also like to thank sir for providing us the opportunity to implement our knowledge on time series.

Abstract

In the following paper we discuss a method of valuing stock options using time series. This may be used as an alternative to what Black-Scholes model is used for, it is a model of valuing option prices, and has been used historically by investors as a trustworthy tool for valuing stocks.

We demonstrate a statistical approach using time series analysis by finding proper order of ARIMA model with help of ACF and PACF plot. This is a more elaborate approach than other models. Instead of computing probabilities of the future prices, and assigning an option value, and summing over all possible outcomes, as done in the formula based approach of Black-Scholes, we focus our model on the first part. That is, rigorous computation and estimation of a series of outcomes and their probabilities to estimate the future option price. We reduce the process of computing option price to simply time series forecasting.

Our project starts off by extracting data from **add data source**, and we normalize the data for better analysis of Option price data. We followed Z-score normalization of the data. Visualisation of the data is an essential step as it provides information about linearity trends and variance of the data, and generating a forecast without plotting the data may spring out erroneous results. This was followed by manipulating the time series, as we decomposed it into constituent seasonal and trend component. We checked the stationarity of the time series using Dickey Fuller test using an inbuilt function in python statsmodel library, adfuller(), which tests the stationarity of the given series and compares the result to a null hypothesis, which is rejected when the process is stationary for certain values elaborated in the *Methodology* section below. If the series is not stationary, which in our case wasn't, we make it stationary with linear differencing.
In the following steps we obtain the ACF and PACF plots to determine the appropriate order of the ARIMA model.

Review of Literature

AR Process

A p^{th} order autoregressive process, denoted AR(p), satisfies:

$$X_t = \theta_1 X_{t-1} + \theta_2 X_{t-2} + ... + \theta_p X_{t-p} + \epsilon_t$$

where ϵ_t is a stationary white noise process and $\theta_p \neq 0$

MA Process:

A q_{th} order moving averages process, denoted MA(q), is characterized by the following equation:

$$X_t = \epsilon_t + \theta_1 \epsilon_{t-1} + \theta_2 \epsilon_{t-2} + \ + \theta_q \epsilon_{t-q}$$

where ϵ_t is a stationary white noise process and $\theta_q \neq 0$.

ARIMA:

ARIMA model is a statistical analysis model that uses time series data to either better understand the data set or to predict future trends. Given a time series X_t, it is said to follow an Auto Regressive Integrated Moving Average(ARIMA) model of order (p, d, q) if

$$Z_t = \Delta X_t = (1-B)^d X_t \ ARMA(p,q)$$

there is, $Z_t = \phi_1 Z_{t-1} + \phi_2 Z_{t-2} ... + \phi_p Zt - p + \epsilon_t + \theta_1 \epsilon_{t-1} + ... + \theta_q \epsilon_{t-q}$ where $\phi_1, ...\phi_p, \theta_1, ..., \theta_q$ are the parameters with $\phi_p, \theta_q \neq 0$ and $\epsilon_t, \epsilon_{t-1}, ...$ are white noise terms.

Options:

An option is derivative in financial market which acts as an insurance, it gives its owner a right to make transaction of an underlying asset.
Option pricing models and tools are very popular tools in the stock market, it is still an area of academic research and widely used by traders on desk, for whom it is a profit making tool. Options are of basically two types:

- **Call Option:** A call option gives the buyer the right but not the obligation to buy the underlying asset at a particular price (strike price) on or before the expiration date.
- **Put Option:** A put option gives the buyer the right but not the obligation to sell the underlying asset at a particular price (strike price) on or before the expiration date.

Option Pricing:

The option pricing concept estimates a price of an alternatives settlement via assigning a cost, called a premium, primarily based totally on the calculated chance that the payment will end in the money (ITM) at expiration.Models used to price options account for variables such as current market price, strike price, volatility, interest rate, and time to expiration to theoretically value an option. Some commonly used models to value options:

- Black-Scholes Model
- Binomial Option Pricing
- Monte-Carlo Simulation

Black-Scholes formula:

The most well known tool in mathematical finance for option pricing for a standard call option is Black-Scholes formula. The formula of Black Scholes Model is given by:

$$C = S_t N(d_1) - Ke^{-rt} N(d_2)$$

where:

$$d_1 = \frac{\ln \frac{S_t}{K} + (r + \frac{\sigma_t^2}{2})}{\sigma_s \sqrt{t}} \qquad d_2 = d_1 - \sigma_s \sqrt{t}$$

where:
C = Call Option Price
S = Stock Price (Current)
K = Strike Price
r = Risk Free Interest-Rate
t = Time of Maturity
N = A Normal Distribution
The Black-Scholes is not that effective because it takes several assumption. The assumptions that Black-Scholes make are:

- No dividends are paid out during the life of the option.
- Markets are random (i.e., market movements cannot be predicted).
- There are no transaction costs in buying the option.
- The risk-free rate and volatility of the underlying asset are known and constant
- The returns on the underlying asset are log-normally distributed
- The option is European and can only be exercised at expiration

The Drawbacks of Black-Scholes model were that it can be only used for European Options i.e. the option cannot be exercised before expiration date. Moreover it assumes that the volatility remains constant, which is not the case in real life scenario.

Stationarity:

A time series is said to be stationary if neither it's expectation nor it's autocovariances are dependent on time. A stationary process exhibits mean reverting behavior, the process tends to remain near or tends to return over time to the mean value.

Augmented Dickey Fuller test:

Augmented Dickey Fuller test is a common practice while checking for stationarity of a time series. As taught in the lectures, in ARIMA(Auto Regressive Integrated Mean Average) time series forecasting, we need to compute the number of differencing that is required to make the time series stationary. There is hypothesis testing involved in Augmented Dickey Fuller test, with an alternate and null hypothesis, and thus there are p-values that are reported. It is a unit root test that tests null hypothesis where $\alpha = 1$ in the following equation.

$$y_t = c + \beta t + \alpha y_{t-1} + \phi \Delta Y_{t-1} + e_t$$

Data Sources

For the calibration of asset pricing models, option pricing data is required. In general we cannot extract historical option prices in a desired format from standard tools, such as Bloomberg terminal. An instance of syntax for locating an expired option:

TICKER: MM/YY OPTION TYPE <C> or <P> STRIKE <Yellow Key> <GO>

Thus, we must provide the following details to find a specific expired option.
(i) Month and Year of expiration
(ii) Type of call option (Call or put)
(iii) Strike
Because of the huge diversity of historical options data and their relatively few applications, no open source dataset exists so for our purpose we obtained the data from Kaggle. The starting date of data is 1st July 2015 and the end date of data is 5th september 2017.

Methodology

This is a step-wise description of the methods followed to produce option price predictions.

Step 0

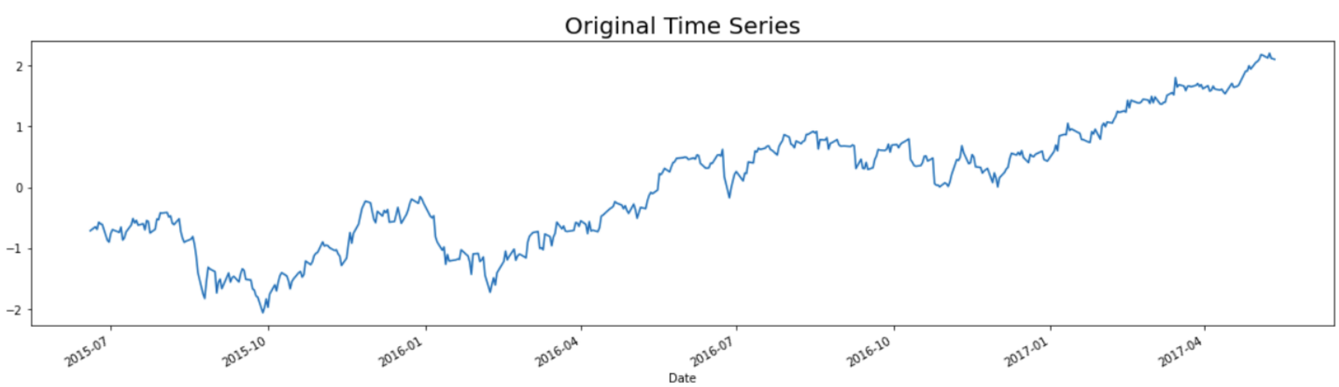
Normalization of data
Normalization is essential for better analysis of Option prices data. We have followed **Z-score** normalization of the data, which infers to the process of normalizing every value in an dataset such mean of all values is 0 and the standard deviation is 1.
i.e.

$$Z - score : X' = \frac{X - \mu}{\sigma}$$

This was followed by an efficient visualization of the data.

Step 1

Visualization of the data
This step gives the information of appropriate weightings, linearity, trend and variance of the data. As the saying goes, generating a forecast without plotting a data, however trivial is a statistical malpractice. Since many a times even sound statistical models produce illogical results, reason being wrong parameters, or misinterpretation of data types and myriad of other trivial look-overs.

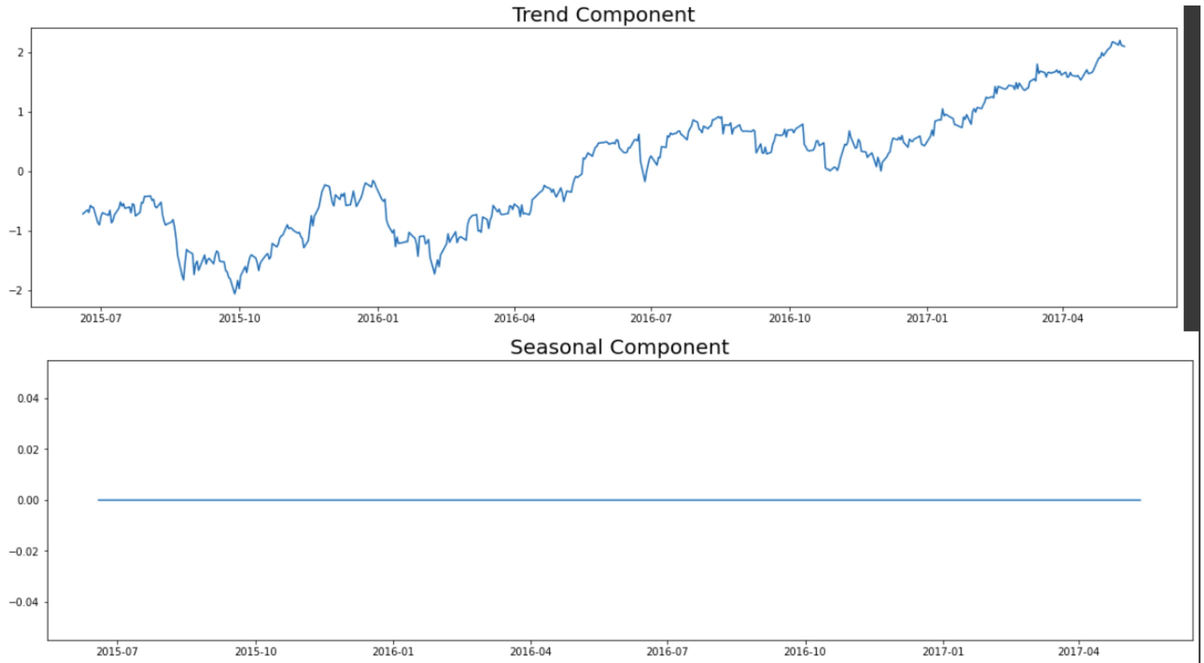


Step 2

Decomposition of time series
We have used *seasonal-decompose* from the *sm.tsa* library. which takes the arguments,

{time series, model type, frequency}

The decomposition is as follows.



Moreover we screened the outliers, which is a data point in the dataset that differs from the observations significantly. Although, outliers in time series may indicate salient features, we rather ignore them for having no predictive value.

Step 3a

Checking for stationarity of time series using ADF testing
In order to determine if the time series process is stationary, we use a type of unit root test, we use the **Dickey Fuller test** using a function present in statsmodel library in Python. Unit roots cause non-stationarity. The Null and Alternate hypothesis of the ADF test are:

- Null hypothesis - Unit root is present i.e. process is not stationary.
- Alternate hypothesis - Unit root does not exist, stationarity exists.

First we import the *statsmodel* package which has a reliable implementation of the Dickey Fuller test in *statsmodels.tsa.stattools*. The function $adfuller()$ returns a tuple of statistics from the Dickey Fuller test, such as the following,

{p-value, test statistic value, lags considered, critical cutoffs}

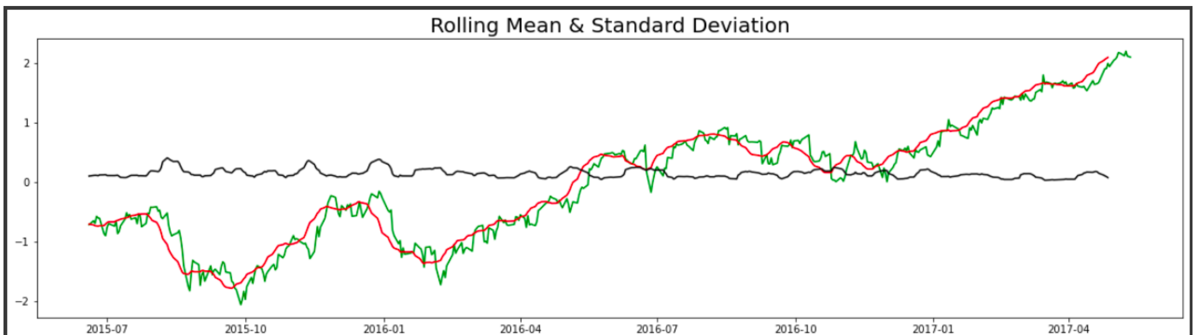
The null hypothesis is rejected and we consider the time series as stationary when,

$$\{\text{test statistic} < \text{critical value}\}$$

With the packages and data entered we can perform the Dickey Fuller using *adfuller*(). The arguments entered are: {arraylike, autolag, maxlag, regression, store, regresults}
We can specify the maximum number of lags with *maxlags* parameter, or let the algorithm decide by itself using *autolag = 'AIC'*. The function would then choose the lag which yields lowest 'AIC'.

```
def test_stationarity(timeseries, window = 12, cutoff = 0.01):

    rolmean = timeseries.rolling(window).mean()
    rolstd = timeseries.rolling(window).std()

    fig = pyplot.figure(figsize=(20, 5))
    orig = pyplot.plot(timeseries, color='green',label='Original')
    mean = pyplot.plot(rolmean, color='red', label='Rolling Mean')
    std = pyplot.plot(rolstd, color='black', label = 'Rolling Std')
    pyplot.title('Rolling Mean & Standard Deviation',fontsize=20)
    pyplot.show()

    print('Dickey-Fuller Test:')
    test = adfuller(timeseries, autolag='AIC', maxlag = 20 )
    output = pd.Series(test[0:4], index=['Test Statistic','p-value','#Lags Used','Number of Observations Used'])
    for key,value in test[4].items():
        output['Critical Value (%s)'%key] = value
    pvalue = test[1]
    if pvalue < cutoff:
        print('p-value = %.5f. The series is stationary.' % pvalue)
    else:
        print('p-value = %.5f. The series is non-stationary.' % pvalue)

    print(output)
```

Results:




```
Dickey-Fuller Test:
p-value = 0.2761. The series is non-stationary.
Test Statistic                  -2.024031
p-value                          0.276137
#Lags Used                       1.000000
Number of Observations Used    477.000000
Critical Value (1%)             -3.444134
Critical Value (5%)             -2.867618
Critical Value (10%)            -2.570008
dtype: float64
```

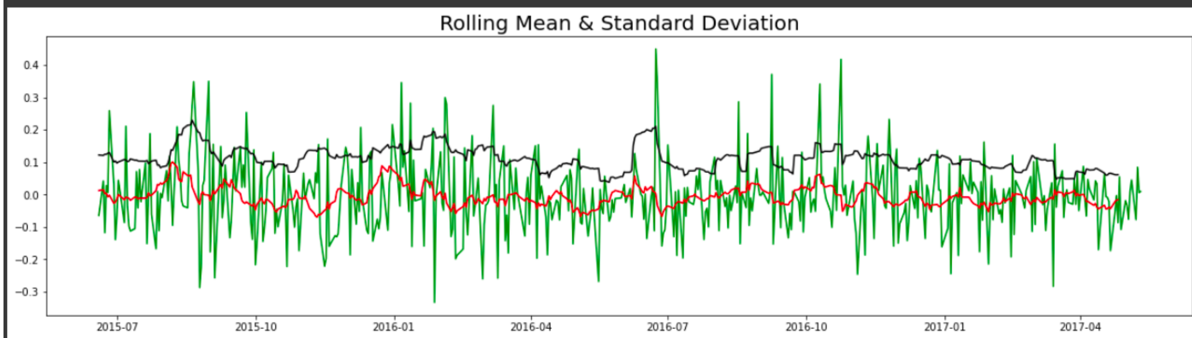
Since the ADF statistic is more than the critical values and p-value obtained is higher than 0.05, the null hypothesis is accepted and we infer that the time-series process is non-stationary. Thus, we move onto the next step, we make the given time series stationary using method of differencing.

Step 3b

Making the time series stationary
We kept on differencing the given time series till the ADF test null hypothesis is rejected. The time series was made stationary just by first-differencing. Moreover, on plotting the decomposition of the time series, there is no seasonal component present in the original time series. A stationary time series has a constant mean, autocovariance and variance. Indicators of stationary time series.

(i) Constant mean.
(ii) Constant variance.
(iii) ACF(Autocorrelation Function) decays fast.
(iv) On performing the Dickey-Fuller Unit Root test a low p-value is given out.

```
first_diff = series - series.shift(1)
first_diff = first_diff.dropna(inplace = False)
test_stationarity(first_diff, window = 12)
```




```
Dickey-Fuller Test:
p-value = 0.0000. The series is stationary.
Test Statistic                 -23.553575
p-value                          0.000000
#Lags Used                       0.000000
Number of Observations Used    477.000000
Critical Value (1%)             -3.444134
Critical Value (5%)             -2.867618
Critical Value (10%)            -2.570008
dtype: float64
```

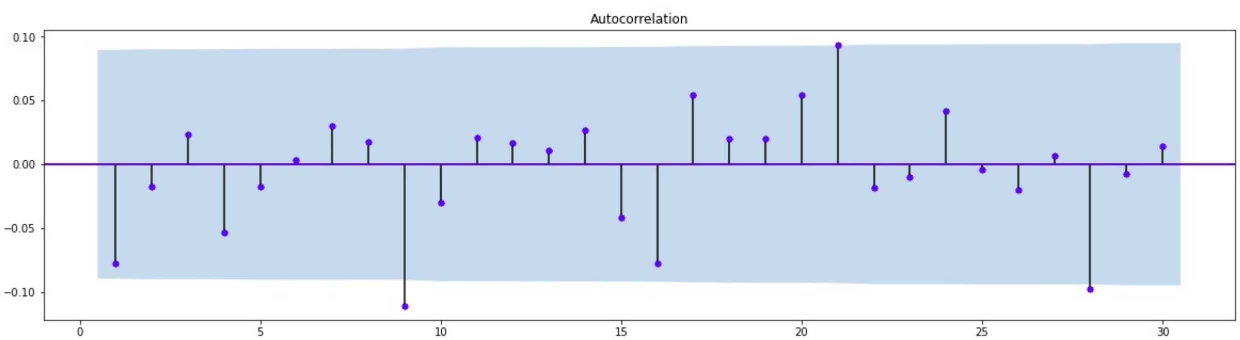
Step 4

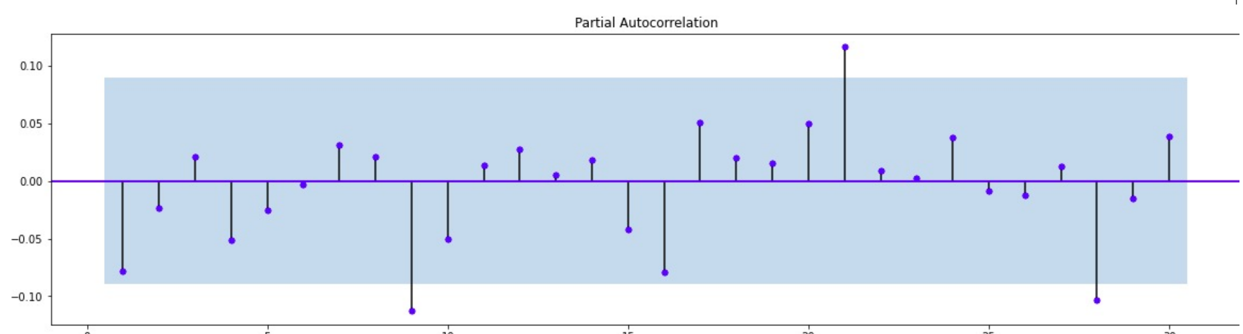
ACF and PACF plots

Finding and interpretation of ACF(AutoCorrelation Function) and PACF(Partial AutoCorrelation Function) is an essential step before we move on to the next crucial step which is fitting an ARIMA model. ACF and PACF facilitate the order of ARMA model, they provide an idea of which model can be a good fit for the given time series. It explains how the current values of the given time series are correlated with the past values. ACF plot is a bar chart, of coeffecient of correlations between the given time series.

Partial AutoCorrelation Function or PACF explains the partial correlation between the series and lags of itself. It can be explained using a linear regression where we predict y_t from y_{t-1}.

We used the *plot-acf* and *plot-pacf* functions from the *statsmodels* library in python. The resultant plots are:





Step 5

Find ARIMA order and building the model

Having made the given time series stationary we next try to fit an ARIMA model with appropriate parameters, to remove any autocorrelation that may be present in the differenced series. We had to try various orders of parameters which was handled by the *model.fit* package in python. By looking at the ACF and PACF plots of differenced series we identified the number of AR and MA terms that are needed. The ARIMA series with orders as

$$\text{ARIMA}(\text{p} = 1, \text{d} = 1, \text{q} = 1)$$

fits the given time series the best.

```
                              ARIMA Model Results
==============================================================================
Dep. Variable:                D.Close   No. Observations:                  478
Model:                 ARIMA(1, 1, 1)   Log Likelihood                 357.455
Method:                       css-mle   S.D. of innovations              0.115
Date:                Wed, 17 Nov 2021   AIC                           -706.910
Time:                        20:27:55   BIC                           -690.231
Sample:                             1   HQIC                          -700.353

==============================================================================
```

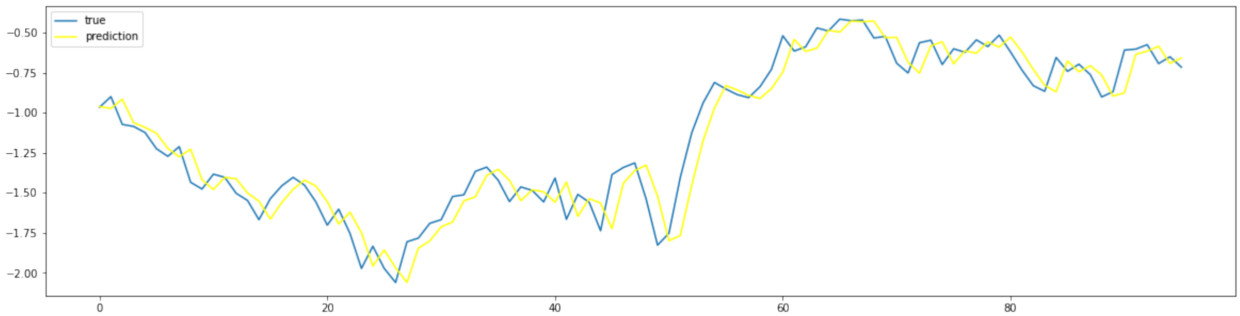
Step 6

Prediction of Option prices

Now we perform forecasting by dividing data into two parts. We use 80% data to train the model and 20% of the data set to test the trained model. Finally we plot the true vs predicted graph.

Conclusions and Main findings

After doing all analysis and predicting option price, We conclude that our model prediction closely resembles the actual option prices. Therefore our model turns out to be more effective than Black-Scholes and Binomial Model.



Trading and Real Market application

We can use some option trading strategies if we can predict the trend of market. Some Risk Optimized trading strategies are:

- **Bear Call Spread:** A bear call spread is achieved by purchasing call options at a specific strike price while also selling the same number of calls with the same expiration date, but at a lower strike price. If the trader believes the underlying stock or security will fall by a limited amount between the trade date and the expiration date then a bear call spread could be an ideal play.

- **Bull Put Spread:** The strategy employs two put options to form a range, consisting of a high strike price and a low strike price. A bull put spread is an options strategy that an investor uses when they expect a moderate rise in the price of the underlying asset.

- **Bull Call Spread:** The strategy uses two call options to create a range consisting of a lower strike price and an upper strike price. A bull call spread is an options trading strategy designed to benefit from a stock's limited increase in price.

- **Bear Put Spread:** A bear put spread is achieved by purchasing put options while also selling the same number of puts on the same asset with the same expiration date at a lower strike price.A bear put spread is a type of options strategy where an investor or trader expects a moderate-to-large decline in the price of a security or asset and wants to reduce the cost of holding the option trade.

- **Iron Butterfly:**A short call and put are both sold at the middle strike price, which forms the "body" of the butterfly, and a call and put are purchased above and below the middle strike price, respectively, to form the "wings." Market players use this strategy during times of lower volatility, when they believe the underlying instrument will stay within a given price range through the options' expiration date.