True Beacon (Quant Research Assignment)

To examine the hypothesis, we devised a pairs trading strategy aimed at capturing any volatility dispersion between two indices. To implement this model, we opted for the Python language, utilizing functions to introduce modularity for specific tasks and repetitive processes.

In summary:

1. We imported Parquet data into a Pandas Dataframe.
2. Constructed a foundational Z-Score model using specified spread and PnL equations.
3. Developed several machine learning and regression algorithms with the goal of improving model evaluation.
4. Conducted a comprehensive comparison of all proposed models against the baseline model.

Detailed Walkthrough:

Data:

• The Parquet file comprises minute-level data spanning from January 1, 2021, to June 30, 2022.

• Total minutes of data: 690,512.

• Trade Time Filtering (9:15 AM - 3:30 PM): 180,856 minutes (481 days).

We created 376-minute blocks for each day, excluding every other minute to reduce noise. (180,856 / 376 = 481 days)

• Saturday-Sunday Filtering: 129,720 minutes (345 days).

Days of the week were assessed, and Saturdays and Sundays were omitted due to their insignificance. (129,720 / 376 = 345 days)

• Non-Traded Days Filtering (No data for the day): 118,440 minutes (315 days).

We utilized a logic check to identify consecutive 376 minutes on a given date with a consistent spread from 9:15 AM to 3:30 PM. Such instances were removed, assuming they represent trading holidays or days with no available data. (118,440 / 376 = 315 days)

Upon closer examination of the data, we observed that TTE (Time-to-Expiry) values remain constant on non-traded days and are irrelevant to the PnL formula. Additionally, there are two values for TTE on the same day for different minutes. To address this noise, we opted to consider the TTE value at 9:15 AM as representative for the entire day.

Instances of Multiple TTE Values on the Same Day:

2021-01-25 2021-02-22 2021-02-23 2021-04-26 2021-04-27

2021-05-24 2021-05-25 2021-07-26 2021-07-27 2021-09-28

2021-10-25 2021-10-26 2022-01-24 2022-03-28 2022-03-29

2022-04-25 2022-04-26 2022-06-27 2022-06-28

- After applying our data cleaning logic to the entire dataset, we further process it by utilizing forward fill to replace any missing values in the dataset.

Z-Score Base Model:

This base_model function calculates rolling mean, rolling standard deviation, and Z-score based on the spread data. It then generates long and short signals using these metrics and applies the timeframe and evaluation functions. The Z-scores' statistics are printed for analysis.

Timeframe function:

The timeframe function manages various time-related aspects of the dataset, including position holding time, signal types, and positions. It calculates the cumulative profit and loss (PnL) based on the spread, time to expiration (tte), and position information. The function ensures proper handling of position transitions, timeframes, and updates relevant columns to facilitate further analysis of the trading strategy's performance over time.

If we find any signal we enter a position and we stay in it for atleast 30 minutes despite having an opposite signal meanwhile and after 30 minutes, if there is no opposite signal we try to hold it till maximum of 1880 minutes(5 days of trade hold).

Evaluation function:

The evaluation function evaluates the performance metrics of a specific trading model.

- It takes calculated PnL (Profit and Loss) values from the dataset and computes essential metrics.
- The function calculates the Sharpe Ratio, providing insights into the risk-adjusted returns of the trading strategy.
- It computes the total PnL, representing the cumulative profit or loss generated by the model.
- The maximum drawdown is determined, measuring the largest drop in cumulative PnL from a peak to a trough.
- The length of the dataset is recorded, indicating the duration or number of data points considered in the evaluation.
- The results are appended to a global report, allowing for a comprehensive comparison of multiple models.
- In essence, the evaluation function provides a standardized approach to assess and compare the effectiveness of different trading models, considering both profitability and risk metrics.

$$\text{Spread} = \text{Bank Nifty IV} - \text{Nifty IV}$$

$$\text{P/L} = \text{Spread} \times (\text{Time To Expiry})^{0.7}$$

According to the PnL formula, a loss is incurred only when the spread is negative. After scrutinizing the dataset, it becomes evident that the spread is exclusively negative on the date 25/10/2022. Consequently, any drawdown or loss would occur solely on that specific day, contingent upon an open position during that period.

Alternative Models:

We explored various models, including Linear Regression, Gradient Boosting, Random Forest Regressor, and Support Vector Machine. In this process, we generated lagged features, addressed missing values using appropriate methods, created a target variable 'y' representing future spread, and prepared features 'X'. Subsequently, we divided the data into training and testing sets, typically using a 70% training and 30% testing ratio (adjustable as needed). Hyperparameters were defined and models were initialized, trained, and predictions were made. A dynamic threshold was established using a rolling mean, and the threshold value was calculated as the dynamic set multiplied by a predetermined factor.

Long signals were generated if the predicted spread exceeded the threshold, while short signals were generated if the predicted spread was below the negative threshold. The data was then processed through the 'timeframe' function, which analyzed trade positions, holding durations, and cutting points. Finally, based on these cumulative results, an evaluation of the model's performance was conducted.

Deliverable #1:

A Z-Score pair trading system is a statistical arbitrage strategy that seeks to identify and exploit deviations in the spread between two related assets or financial instruments. The Z-Score is a measure of how far a particular data point is from the mean of a group of data points, expressed in terms of standard deviations. In pair trading, it is often used to assess the relative value of the spread between two assets. The rationale behind Z-Score pair trading is that it assumes the spread between the two assets will eventually revert to its historical mean.

After cleaning and preprocessing the data, establishing the base model, and generating long and short signals based on predefined formulas, we proceed to assess entry and exit points. Finally, as we delve into the numerical analysis phase to evaluate its performance, presented below are the results of the Z-Score-based pair trading system.

Z-score model performance:

	Model	Sharpe Ratio	Total PnL	Max Drawdown
0	Base Model	1.751863	59534.975777	0.076773

Deliverable #2

Linear Regression:

- Linear regression is a simple and widely used supervised learning algorithm for predicting a continuous outcome based on linear relationships between input features and the target variable.
- It assumes a linear connection between the features and the response, aiming to find the best-fitting line through the data points.
- The model minimizes the sum of squared differences between predicted and actual values to find optimal coefficients.

Gradient Boosting:

- Gradient boosting is an ensemble learning technique that combines weak predictive models, typically decision trees, to create a strong predictive model.
- It builds trees sequentially, with each tree correcting the errors of the previous ones.
- The model minimizes a loss function by adjusting weights for misclassified instances during training.

Random Forest:

- Random Forest is an ensemble learning method that constructs multiple decision trees during training and outputs the average prediction (regression) or majority vote (classification) of the individual trees.
- It uses bootstrapped sampling and random feature selection to introduce diversity among the trees, reducing overfitting.
- The final prediction is an ensemble average, enhancing model robustness.

Support Vector Machine (SVM):

- SVM is a supervised learning algorithm used for both classification and regression tasks.
- It works by finding the hyperplane that best separates different classes in the feature space.
- SVM aims to maximize the margin between classes, making it robust to outliers.
- The algorithm can be adapted for nonlinear problems using kernel functions.

Deliverable #3:

Model Comparison and Evaluation

	Model	Sharpe Ratio	Total PnL	Max Drawdown
0	Base Model	1.751863	59534.975777	0.076773
1	Base Model	0.517156	12626.177655	0.000000
2	Linear Regression	0.517474	12635.334750	0.000000
3	Gradient Boosting	0.517489	12635.671932	0.000000
4	Random Forest	0.517472	12635.232011	0.000000
5	SVM	0.517191	12627.162922	0.000000

We have showcased the assessment outcomes for the foundational model, drawing a direct parallel with diverse models. This comparative analysis is grounded in a uniform 70% training and 30% testing dataset distribution, with key metrics including Sharpe ratio, total PnL, and maximum drawdown serving as benchmarks for performance evaluation.

So with the goal of optimizing the absolute PnL, Sharpe Ratio and Drawdown of the strategy, we can keep Gradient Boosting Model as our Proposed Model.

The results are contingent on the assumptions that rolling parameter changes are considered accurate when accounting for the formulas, entry and exits. It's important to note that the presented performance is without the consideration of any transaction costs, slippages, changes in market regime, that may occur with uncertainity.