REPORT ON TRUE_BECON ASSIGNMENT CODE

March 12 2024

Shubhankar Mondal

Mtech AI’25

IISc Bangalore

1. Project background and description

The provided code focuses on analyzing financial data using statistical models, specifically implementing a Z-Score model and an enhanced Random Forest model

2. Data Loading and Preprocessing:

- loaded the data from a Parquet file using `pandas.read_parquet`.
- Performed data cleaning by handling missing values with `df.dropna()` and filtering the data between '09:15' and '15:30' using `.between_time`.
- Checked data types using a custom function `check_data_types`.

3. Spread and P/L Calculation

- Defined a spread column by subtracting the `nifty` price from the `banknifty` price (`df['spread'] = df['banknifty'] - df['nifty']`)

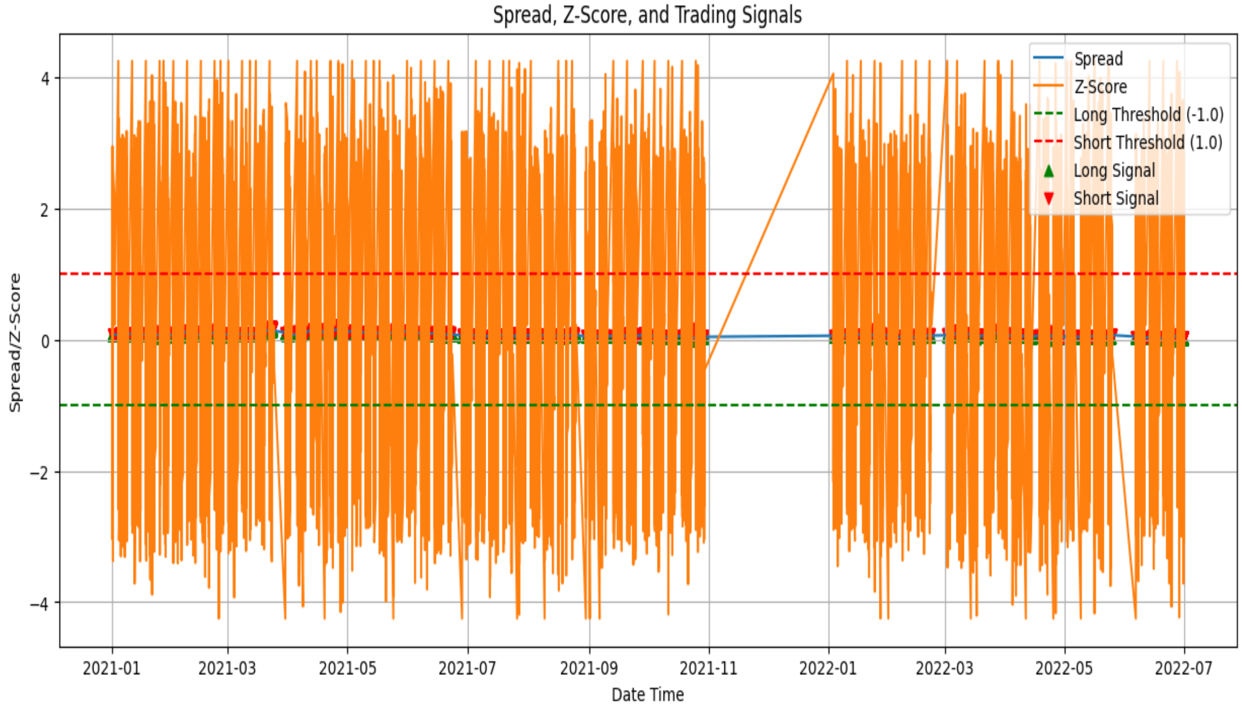
  .

- Defined a function `calculate_pl` to calculate the Profit/Loss (P/L) based on spread difference, position, and Time To Expiry (TTE).

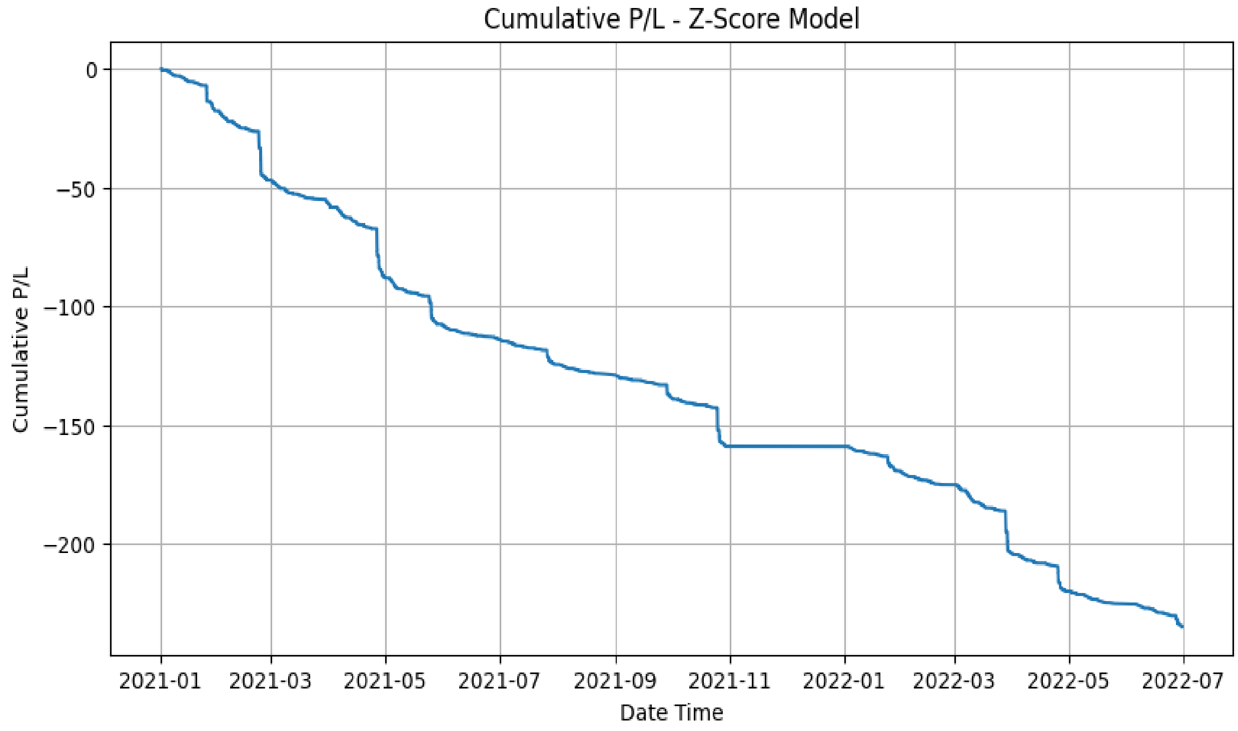
4. Z-Score Model

- Implemented a Z-Score model for identifying entry and exit signals:
  - Calculated the Z-score using a rolling window mean and standard deviation (`calculate_zscore`).
  - Defined thresholds (`threshold_zscore_long` and `threshold_zscore_short`) for long and short positions based on the Z-score.
  - Created a function `calculate_zscore_position` to generate trading signals based on the Z-score thresholds.
  - Visualized the spread, Z-score, and trading signals using `matplotlib.pyplot`.
  - Calculated the daily P/L for the Z-score model and plotted the cumulative P/L.

5. Plots for Z -Scored Based Model:



Spread, Z-Score, and Trading Signals



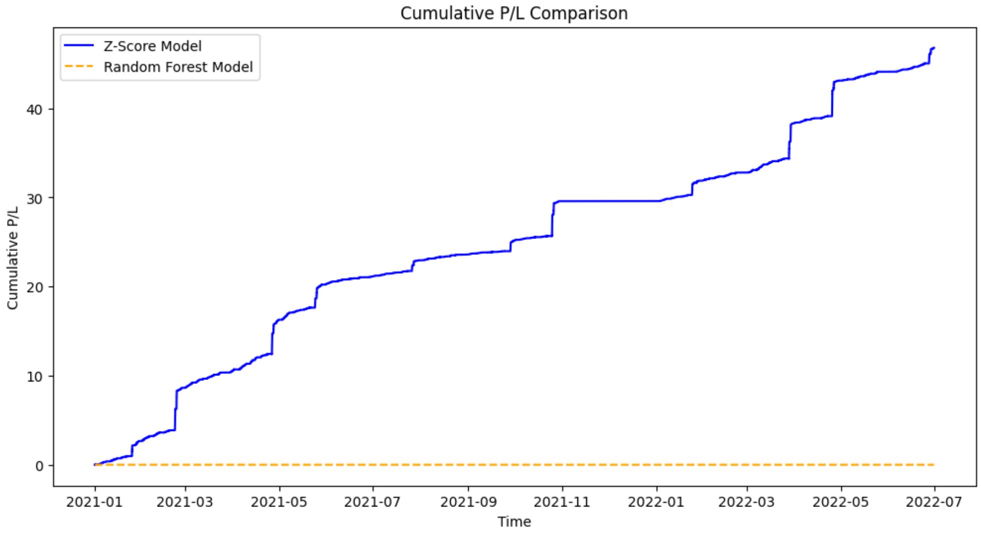
Cumulative P/L - Z-Score Model

6. Random Forest Model:

- Implemented a Random Forest Regression model for spread prediction:
  - Performed feature engineering by calculating moving average, standard deviation of spread, and squared TTE (`df_rf['spread_ma']`, `df_rf['spread_std']`, and `df_rf['tte_squared']`).
  - Split the data into training and testing sets using `train_test_split`.
  - Defined a function `train_evaluate_rf` to train the Random Forest model and evaluate its performance using Mean Squared Error (MSE).
  - Created a function `calculate_pl_random_forest` to calculate the P/L based on the model's predicted spread and defined thresholds (`threshold_long_rf` and `threshold_short_rf`).

7.

8. Plots for Random Forest Based Model:



9. Metric Comparisons:

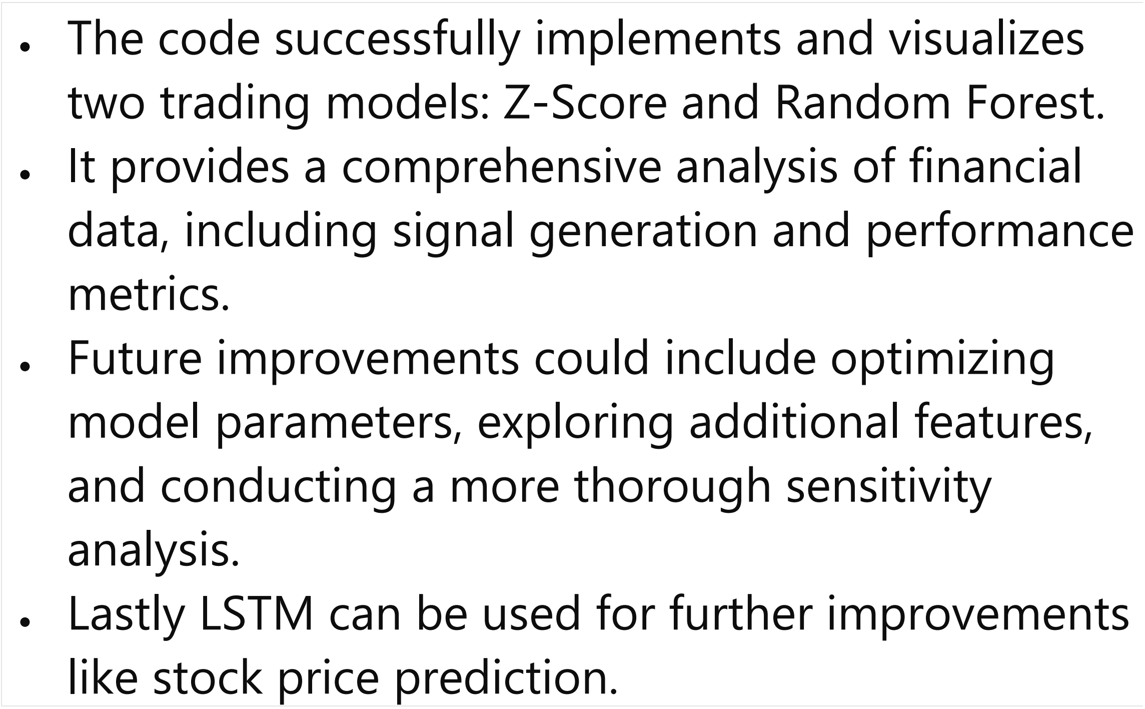
```
Base Z-Score Model:
Total P/L:
46.794352849999996
Sharpe Ratio:
0.12428119394963918
Max Drawdown: -
0.053550499999999295
```

```
Enhanced Random
Forest Model:
Total P/L: 0.0
Sharpe Ratio: nan
Max Drawdown: 0.0
```

10. Performance Evaluation:

- Defined a function `calculate_performance_metrics` to calculate total P/L, Sharpe Ratio (a measure of risk-adjusted return), and maximum drawdown for each model.
- Compared the performance metrics (total P/L, Sharpe Ratio, and maximum drawdown) between the Z-score and Random Forest models.
- Visualized the cumulative P/L of both models for comparison.

11. Summary and Recommendations:



- The code successfully implements and visualizes two trading models: Z-Score and Random Forest.
- It provides a comprehensive analysis of financial data, including signal generation and performance metrics.
- Future improvements could include optimizing model parameters, exploring additional features, and conducting a more thorough sensitivity analysis.
- Lastly LSTM can be used for further improvements like stock price prediction.