WARWICK BUSINESS SCHOOL
THE UNIVERSITY OF WARWICK

# Assignment 3: Machine learning and AI for intraday return prediction

# Data download

- **Same as in assignment 2:**
- Download already cleaned and merged trades and quotes data (from Daily TAQ database)
  - Company: Citigroup, ticker “C”
  - Time period: Jan 1, 2024 - Jan 31, 2024
  - Step_3_C_Trades_Quote_Joined.parquet
- Import Parquet file into Pandas dataframe

# Defining variables

- **Same as in assignment 2:**
- Calculate following variables for each minute:
  - average relative quoted spread
  - order imbalance
  - average depth imbalance as (ASKSIZ-BIDSIZ)/(ASKSIZ+BIDSIZ)
  - traded volume
  - one-minute return from closing mid-prices for each minute
  - realized volatility as sum of squared 1-minute returns over the past hour
- Winsorize all variables at 1% and 99%

# Random forest with walk-forward testing

- Use the same walk-forward strategy as in assignment 2, but replace OLS with **Random Forest** estimations (refitted daily):
  - Train a model on a 2-week window
  - Predict one-minute returns for next day
  - Slide the window one day forward and repeat
  - Hint: Think carefully about parameters “n_estimators” and “min_samples_leaf”!

# Random forest with walk-forward testing

- Store
  - feature importances from each refit
  - predicted and actual returns for each minute
- Recalculate $R^2$, RMSE, MAE, correlation and compare to corresponding OLS metrics
- Calculate average feature importance over all refits and compare to corresponding OLS estimations

# Backtesting: Random forest

- Backtest RF walk-forward strategy (including transaction costs!):
  - compute total cumulative return and plot it on the graph
  - average daily return, std of daily returns, daily Sharpe ratio
  - hit rate, max drawdown, turnover
  - t-stats of strategy daily returns
- **Question: Compare the strategy performance between RF and OLS (with transaction costs). Would you trade based on your strategy? Why or why not?**

# XGBoost with walk-forward testing

- Use the same walk-forward strategy, but replace Random Forest with **XGBoost** estimations (refitted daily)
  - Hint: Think carefully about following parameters:
    - n_estimators, max_depth, learning_rate, min_child_weight
- Recalculate $R^2$, RMSE, MAE, correlation and compare to corresponding OLS/RF metrics
- Calculate average feature importance over all refits and compare to corresponding OLS/RF estimations

# Backtesting: XGBoost

- Backtest XGBoost walk-forward strategy (including transaction costs!):
  - compute total cumulative return and plot it on the graph
  - average daily return, std of daily returns, daily Sharpe ratio
  - hit rate, max drawdown, turnover
  - t-stats of strategy daily returns
- **Question: Compare the strategy performance between XGBoost, RF and OLS (with transaction costs). Which model would you choose: OLS, RF or XGBoost? Why?**

# Evaluating News Sentiment with FinBERT

- Scrape Google News RSS for Citigroup and filter strictly for January 2024:
  - Example for a news search for climate change:
    - `https://news.google.com/rss/search?q=climate+change`
  - Make sure to search across several keywors, e.g. 'Citigroup', 'Citi', 'Citibank', 'C', 'C stock'
  - Parse RSS feed (extract “title”, “timestamp” and “link”)
  - Save in dataframe “news_items”

# Evaluating News Sentiment with FinBERT

- Use FinBERT sentiment analysis model

  ```
  “import torch
  from transformers import pipeline
  sentiment = pipeline("sentiment-analysis",
  model="ProsusAI/finbert")”
  ```

- Apply FinBERT sentiment model to the title column and store **sentiment label**
- Map sentiment label to numeric sentiment signal:
  - “positive” = 1; “neutral” = 0; “negative” = -1
- Compute average sentiment signal for each day

# Random Forest with “Sentiment signal”

- Merge daily news sentiment to existing TAQ data
- Re-estimate the Random Forest model with "Sentiment_signal" included as an additional feature
- Backtest the Random Forest model with "Sentiment_signal" (with transaction costs)
- **Question: Compare the strategy performance between RF with and without sentiment signal. Which model would you choose? Why?**