DADS6003 Final Project
Stock Price Buy/Sell Classification

Thunchanok	Surasunsanee	6520422007
Budsadee	Sareerasart	6520422009
Therarat	Srisaswatakul	6520422012
Pacharapon	Pukpiboon	6520422013
Thanyakorn	Hovongratana	6520422018

Prediction of Stock Price direction

- **Introduction**
  - **Objective**
    - To predict whether we should buy or sell 3 stocks (ERW, TISCO and SPRC) according to the following conditions.
      - Time window used for prediction is 15 minutes.
      - Binary classification includes Buy (1) and Sell (-1).
      - Metric to evaluate the model is AUC.
  - **Motivation**
    - We try to create algorithms to guide day-trading using 15-minute intervals to see if it is possible to "beat the market" and gain above average returns by accurately predicting if stock prices will go up or down.
    - The hypothesis is that this is not possible since the machine learning modelling for financial markets has been done exhaustively, with only moderate predictive power. We would like to explore if this hypothesis holds true by trying to create a model of our own.
- **Background**

Stock price prediction with machine learning is a topic with extensive research, in large part due to the possible financial gains that could be achieved if one were to achieve high accuracy on predictions.

Efficient-market hypothesis (EMH) proposed by Eugene Fuma in 1970 contends that predicting future prices to gain an above average returns is not possible since the market either follows a random walk in a weak-form efficiency state, or already incorporates all relevant information in price movements in a strong-form efficiency state. However, many works have refuted EMH to some extent, including Dremen and Berrys 1995 paper on low P/E stocks having outsized returns and A Non-Random Walk Down Wall Street (Lo and MacKinlay 1999) arguing that there is no random walk. (HJELM 2019)

Investopedia–a prominent investment knowledge portal–also suggests that predicting the stock using historical data is not beneficial to short-term traders because they are usually better off following a confirmed trend than predicting one. (Mitchell 2022)

François Chollet, the creator of Keras, wrote in his book Deep Learning with Python that past performance is not a good predictor of future returns and compared it to driving using a rear-view mirror. Some

citations of hedge funds successfully using machine learning to generate above average returns may indicate that the method might actually be feasible. The more plausible explanation is that while the machine learning model may help in their success, the real competitive advantage of these companies is the data that they feed to the model. They most likely have access to private data that has a big impact on stock price movements. This is in line with the underlying mechanism of machine learning: the model is as good as the data you train it on.

Regardless of the many criticisms, if we look into related works that attempt to predict stock prices, one particular machine learning algorithm dominates the field: Long Short-Term Memory (LSTM), a variation of recurrent neural network (RNN) which is capable of learning order dependence in sequence prediction and is well-suited for handling sequential data with long-term dependencies. However, this model is beyond the scope of the course, we cannot use this model for stock price predictions.

With this in mind, this work will attempt to predict stock prices using more well-known and simpler classification algorithms to see how well these algorithms can predict stock prices.

- **Solution**
    - Create additional features to represent technical indicators.
    - Transform the existing features to create more meaningful features such as circular datetime as vectors, dummy variables representing market open and close.
    - Use nested cross-validation to tune hyperparameters and evaluate the model.
        - Split the data into outer loop and inner loop, both using stratified k-fold cross-validation.
        - Train an algorithm to find the best hyperparameters using scikit-learn GridSearchCV.
        - Cross-validate the model using stratified k-fold to obtain average AUC scores.
    - Test the fitted model on a separate test dataset to obtain test AUC scores.
    - Diagnose the model as needed to obtain an optimal model.
    - We repeat the above process to compare the 5 following algorithms.
        - Logistic Regression
        - K Nearest Neighbors Classifier

    - Gaussian Naive Bayes
    - Random Forest
    - Multi-layer Perceptron classifier
- **Experimental Setup and Result**
  - **Data collection**
    - Yahoo Finance API for 3 tickers of interest
      - TISCO.BK - Tisco Financial Group Public Company Limited
      - ERW.BK - The Erawan Group Public Company Limited
      - SPRC.BK - Star Petroleum Refining Public Company Limited
  - **EDA: Graphs**
    - Correlation of features heat map: We found a high correlation value between features as follows.
      - %D and %K (0.90)
      - Lower Band and S_10 (-0.95)
      - Circular_Day_Sine and Is_Friday (-0.77)
      - Circular_Time_Cosine and Is_4 pm (0.84)
      - Circular_Time_Cosine and Circular_Time_Sine (-0.76)
    - Pair plot between features: Some features have linear relationships as follows.
      - %D and %K
      - %K and RSI
      - %D and RSI
      - Lower Band and S_10
      - Circular_Time_Cosine and Circular_Time_Sine
    - Normality distribution of features
      - All of the original features are not normally distributed.
      - Only RSI and %D are not normally distributed for additional features.
  - **Training Process** (*Cross validation*)
    - The training process uses nested k-fold cross-validation with 10-fold for the outer loop and 5-fold for the inner loop.
    - The inner loop is responsible for hyperparameter tuning using the scikit-learn GridSearchCV method.

    - The outer loop is used to evaluate the selected model from GridSearchCV.
- **Evaluation process** (*Test AUC, Mean of CV*)
    - We observe both test area under curve (AUC) and mean cross validation score (mean and SE of CV scores) to evaluate the model. Final model selection considers both Test AUC and Mean of CV.
        - Ideally, we are looking for high scores for both the test score and CV scores with a low difference between the two scores. This indicates that the model has good generalization performance to unseen data.
        - The difference is tested using null-hypothesis test of difference in Test AUC score and mean of CV AUC scores (H0: Test AUC score = Mean of CV AUC scores).
        - If the two scores greatly differ from each other, this means that the model may not generalize well to unseen data or the selected test data may not follow the general trend.
        - Diagnostics include changing periods of training data and test data, changing hyperparameter space, or adjusting cross-validation number of folds.
- **Results**
    - We report mean and standard errors of the CV AUC scores along with test AUC scores for each fitted algorithm for each ticker.
    - We then select the model with the highest mean CV AUC that also yields Test AUC score within 95% confidence interval of CV AUC, assuming the CV AUC scores are normally distributed.

- SPRC.BK

Algorithm	CV Scores (Mean ± SE)	AUC Score
Logistic Regression	0.628±0.014	0.678
K Nearest Neighbors	0.547±0.013	0.566
Gaussian Naive Bayes	0.567±0.015	0.645
Random Forest	0.578±0.012	0.536
Multi-layer Perceptron classifier	0.558±0.019	0.602

- TISCO.BK

Algorithm	CV Scores (Mean ± SE)	AUC Score
Logistic Regression	0.788±0.027	0.822
K Nearest Neighbors	0.715±0.020	0.744
Gaussian Naive Bayes	0.711±0.028	0.791
Random Forest	0.741±0.027	0.740
Multi-layer Perceptron classifier	0.613±0.059	0.768

- ERW.BK

Algorithm	CV Scores (Mean ± SE)	AUC Score
Logistic Regression	0.670±0.033	0.745
K Nearest Neighbors	0.639±0.029	0.635
Gaussian Naive Bayes	0.668±0.027	0.699
Random Forest	0.654±0.023	0.718
Multi-layer Perceptron classifier	0.585±0.046	0.556

        - We select the following fitted model for each ticker as highlighted in the chart.
            - SPRC.BK - Logistic Regression
            - TISCO.BK - Logistic Regression
            - ERW.BK - Logistic Regression
- **Conclusion and Future work**
    - While the models can predict stock price movements to a certain degree, the precision as measured by AUC scores are not very high (67.8% for SPRC, 82.2% for TISCO and 74.5% for ERW) and they are not consistent between each ticker.
    - We do not expect to be able to use the models reliably to predict stock prices, otherwise investors would have already taken advantage of this to gain above-average market returns.
    - Even if one were to find a working algorithm, it may be because the predicted interval has a stable or "predictable" trend. This is evidenced by the fact the scores vary highly between each stock given the same features as predictors.
    - This does not mean that the work has no value as the models can help us process technical indicators and see trends in the stock prices and have a rough guideline on whether the prices would go up or down. The actual investment decision, however, should not be based solely on the predictions of the models.
    - Possible future improvement to this work involves incorporating LSTM as another algorithm to predict performance. As mentioned earlier, the prevalent machine learning algorithm used to predict stock prices is LSTM, and not the algorithms we use in this work. Hence, if we were to comprehensively compare machine learning model performance, it would be appropriate to include LSTM.
- **Appendix**
    - Variable definition
        - Simple Moving Average
            - 10-period moving average of *Close*
        - Relative Strength Index (*RSI*)
            - RSI values above 70 are often considered overbought, suggesting that a security may be overvalued and a potential rebound could occur.
            - RSI values below 30 are often considered oversold, suggesting that a security may be undervalued and a potential rebound could occur.

- Stochastic Oscillator (*%K* and *%D*)
  - %K, which is from the calculation between Close, High and Low price, is used to indicate whether the stock status is overbought or oversold.
    - %K values above 80 are often considered 'Overbought'.
    - %K values below 20 are often considered 'Oversold'.
  - %D is the simple moving average of %K.
- Bollinger Bands (*UpperBand* and *LowerBand*)
  - *UpperBand* is the difference of Close to upper bound of Bollinger Band.
  - *LowerBand* is the difference of Close to lower bound of Bollinger Band.
- Trend Reversal (*Reverse_buy* and *Reverse_sell*)
  - *Reverse_buy* is a dummy variable to indicate that the ticker starts to change from bearish to bullish trend, if current *Close* is higher than *High* of the previous 2 periods.
  - *Reverse_sell* is a dummy variable to indicate that the ticker starts to change from bearish to bullish trend, if current *Close* is lower than *Low* price of previous 2 periods.
- Monday or Friday (*Is_Monday* and *Is_Friday*)
  - *Is_Monday* Dummy variables is to indicate Monday when we assume that the volume on Monday might be lower or higher than other days apart from Friday.
  - *Is_Friday* Dummy variables is to indicate Friday when we are aware that traders are likely to avoid holding its price over the weekend and the volume of buying is lower than other days.
- Day of Week in Circular Format (*Circular_Day_Sine* and *Circular_Day_Cosine*)
  - *Circular_Day_Sine* is a transformation of day of week using the sine function.
  - (*Circular_Day_Cosine, Circular_Day_Sine)* is a vector that represents a coordinate in a two-dimensional space.
  - The coordinates for the 7 days will form a circle

- Time of Day in Circular Format (*Circular_Time_Sine* and *Circular_Time_Cosine*)
  - *Circular_Time_Sine* is a transformation of hour and minute of the time of the day using sine function.
  - *Circular_Time_Cosine* is a transformation of hour and minute of the time of the day using cosine function.
  - (*Circular_Time_Cosine, Circular_Time_Sine)* is a vector that represents a coordinate in a two-dimensional space.
  - All coordinates for time of day roughly form a circle.
- Market Open and Close Hour *(Is_10am and Is_4pm)*
  - *Is_10am* is a dummy variable that indicates a period of time during 10:00 - 10:59 am.
  - *Is_4pm* is a dummy variable that indicates a period of time during 04:00 - 04:59 pm.

References

n.d. Wikipedia. Accessed January 11, 2024. https://www.reddit.com/r/datascience/comments/zg6qao/understanding_machine_learning_for_financial/?rdt=51514.

HJELM, OSCAR. 2019. “Impact of Time Steps on Stock Market Prediction with LSTM.” DiVA portal. https://www.diva-portal.org/smash/get/diva2:1361305/FULLTEXT01.pdf.

Mitchell, Cory. 2022. “Profit Without Predicting the Market - Technical Analysis.” Investopedia. https://www.investopedia.com/articles/trading/10/profit-without-predicting.asp.