AITI: Jurnal Teknologi Informasi, Volume 21 No. 1 Maret 2024, 140-151
ISSN 1693-8348 E-ISSN 2615-7128

JURNAL
TEKNOLOGI INFORMASI

A comparative study on classification models for stock rating prediction

Justin Yap [1], Trianggoro Wiradinata [2]
[1,2]School of Information Technology, Universitas Ciputra Surabaya
CitraLand CBD Boulevard, Made, Kec. Sambikerep, Kota Surabaya
Email: japjustin@yahoo.com

Article History

Received:	Revised:	Disetujui:
06-02-2024	28-02-2024	28-02-2024

Abstract

*The digital transformation in the stockbroker industry has led to a significant increase in retail investors, who often lack the expertise to analyse stocks thoroughly. This research addresses the challenge by proposing a classification model to predict stock ratings such as "Reduce", "Hold", "Moderate Buy", and "Buy", allowing retail investors to make informed decisions. The data analysed is collected from the S&P 500 index through web scraping using Beautiful Soup, resulting in a dataset used for training and testing the classification model. Popular stock indicators are used as attributes in predicting the rating of the stock, which includes the Exchange, Price, Volume, Market Cap, ROE, ROA, P/E Ratio, EPS, Annual Sales, Net Income, Net Margins, and PB Ratio of the stock. The models selected for classification include K-Nearest Neighbors (k-*NN*), Gaussian Naive Bayes, Support Vector Machine (SVM), Decision Tree, and Random Forest. GridSearch is employed to maximize each algorithm's parameters for optimal performance. Results indicate that the k-*NN *model outperforms others, achieving the highest accuracy (0.618644) and weighted F1-score (0.605011). However, all models exhibit relatively low accuracy, suggesting the complexity of predicting stock ratings due to external factors not considered in the study.*

Keywords: *Stock Rating, Machine Learning, Classification, Web Scraping, S&P 500*

Introduction

The amount of retail investors is growing rapidly due to the digital transformation in the stockbroking industry [1]. Retail investors are defined as nonprofessional investors who buy and sell stocks through a brokerage firm. All investors have one goal in mind, that is to gain as much profit as possible. However, when compared with institutional investors, most retail investors lack the knowledge and resources to analyse the potential of a certain stock.

Stock analysis is mainly divided into two types, fundamental analysis and technical analysis. Fundamental analysis involves measuring a company's intrinsic value by analysing the company's earnings, expenses, assets, and liabilities to provide a net present value, while technical analysis focuses on analysing the price chart of the stock itself to determine where the price will be headed. Stock analysts mostly rely on fundamental analysis to determine the long-term valuation of a stock [2]. According to Wnuczak [3], investing based on stock market recommendations can be a profitable strategy in the long run.

However, getting recommendations from stock analysts also means that investors would need to pay for their services. Some brokers do offer the services of stock analysts for free to retail investors. However, they are usually limited to popular stocks, leaving the less popular tickers without any ratings. Thus, a classification model will be created to determine the rating of a stock based on various popular stock indicators.

Over the years, a great deal of research has been carried out that make use of machine learning algorithms, from those that are able to predict music trends [4] to those that can help determine road construction priorities [5]. Machine learning has also been applied in the fields of finance and economics, such as forecasting real estate prices. [6]. Traditionally, stock price prediction models primarily relied on conventional techniques [7]. However, with the advancements in machine learning, there has been a noticeable shift towards leveraging more sophisticated algorithms. These models aim to predict stock prices by analysing a combination of historical data, market indicators, and economic factors. Under different market scenarios, machine learning models like Random Forest classifiers and Support Vector Machine (SVM) have been demonstrated to be helpful in analysing and forecasting stock prices [8], [9]. Many studies aim to forecast future stock prices. However, the accuracy of these predictions often falls short in real-world scenarios, attributed to the frequent fluctuations in stock prices [10].

Classification, a key facet of machine learning, plays a crucial role in predicting stock ratings with a categorical nature. Unlike regression, which predicts continuous variables, classification models focus on assigning predefined labels to data points based on patterns and features. It has been demonstrated that the use of associative classifiers in short-term stock trading increases the explainability of stock rating predictions and produces dependable signals [11].

To evaluate the effectiveness of these classification models, researchers employ metrics such as accuracy and weighted F1 score [12]. These metrics serve as benchmarks for assessing the models' ability to correctly classify stock ratings, providing a comprehensive understanding of their performance. The study of classification models for stock rating prediction represents a significant stride in enhancing the predictive capabilities of financial analysis. By exploring and comparing the effectiveness of various machine learning algorithms, this research aims to contribute to the development of accurate and reliable models for predicting stock ratings based on popular indicators.

Methods

The diagram illustrates the data science methodology, spanning from understanding the business context to evaluating the model (refer to Figure 1). Multiple iterations of parameter tuning were conducted to enhance the performance of the machine learning model.

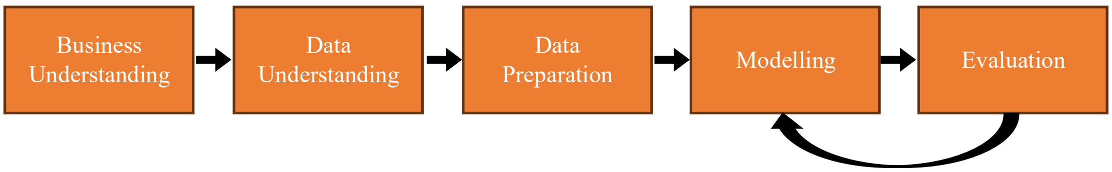

Figure 1 Diagram of the research process

The Target Variable are Rating: Analysts' rating of the stock. There are 4 different values in the Rating column: "Reduce", "Hold", "Moderate Buy", and "Buy". There are features affecting the target variable. Such as: Exchange: The platform for buying and selling the stock, Price: The current value of the stock, etc.

The dataset will be obtained by scraping the https://www.marketbeat.com/ website for the stocks in the S&P 500 index (the most popular stocks from the United States of America) using the Beautiful Soup library for Python. The scraped data will be saved onto a CSV file named "stocks.csv". This dataset will be used to train as well as to test the classification model.

To process the data, the dataset or the "stocks.csv" file must be turned into a DataFrame. Turning the dataset into a DataFrame allows the usage of functions from the Pandas library. Some columns from the DataFrame will also be excluded, which includes the unnamed column consisting of the index numbers of the records, the symbol column, and the industry column. These columns are removed since they do not have any significance to the analysis.

Results and Discussions

After turning the dataset into a DataFrame, pre-cleaning will be performed. Using the head() function previously, it becomes evident that some of the values are missing or stated as NaN. Records with NaN values will be removed using the dropna() function. Along with that, most of the continuous values still contain strings within them such as '%' or '$'. If the info() function is called on the DataFrame, it becomes apparent that a majority of the data features are of type object. The conversion of these continuous columns is necessary so that its values can be used in the analysis. A function, converter(x), will be created to convert strings representing numbers with 'million', 'billion', or 'trillion' suffixes into their numerical equivalents (refer to Program Code 1).

Program Code 1 Defining the converter function

```
def converter(x):
    if 'million' in x:
        return f"{(float(x.strip(' million'))*1000000)}"
    elif 'billion' in x:
        return f"{(float(x.strip(' billion'))*1000000000)}"
    elif 'trillion' in x:
        return f"{(float(x.strip(' trillion'))*1000000000000)}"
    else:
        return x
```

We will then proceed to convert the object type into the appropriate type. Using the replace() function, we can replace certain strings inside of our records. For example, we will be removing the '$' string for values in the Price column. Afterwards, we can convert the column type into float. For columns with values that have the 'million', 'billion', or 'trillion' suffixes, we will apply the converter function using Pandas apply() function before converting them into float type.

For the enhancement of dataset comprehension and the acquisition of valuable insights, Data Visualization will be conducted, specifically focusing on Exploratory Data Analysis (EDA). EDA involves the crucial process of conducting preliminary investigations on data to uncover patterns, identify anomalies, test hypotheses, and validate assumptions using summary statistics and visual representations. The EDA Visualization step will be divided into two categories: EDA (Univariate) and EDA (Multivariate). Univariate analysis involves studying a single variable, while multivariate analysis involves examining multiple variables.

To obtain a concise overview and structure of the dataset, a sample can be extracted using the head() function, providing the first 5 instances of the data. Following that, the data's shape can be determined using the shape() function, which reveals the total number of columns and rows – in this case, 472 rows/records and 13 columns, inclusive of the target variable. The data's features can be categorized into numerical and categorical. Numerical features encompass: Price, Volume, Market Cap, ROE, ROA, P/E Ratio, EPS, Annual Sales, Net Income, Net Margins, and PB Ratio. The two categorical features are Exchange and Rating. Subsequently, a comprehensive exploration and analysis of both the numerical and categorical features within the dataset will be delved into.

The understanding of numerical features in the dataset can be enhanced by employing the describe() function, which provides comprehensive statistical information such as mean, range, standard deviation, and more (refer to Figure 2). Notably, examining the visualizations reveals insights into the data's range, where the range between some features are quite high. Although there are substantial differences in the ranges of these data, normalization will be applied during the data pre-processing stage.

	Price	Volume	Market Cap	ROE	ROA	P/E Ratio	EPS	Ann
count	472.000000	4.720000e+02	4.720000e+02	472.000000	472.000000	472.000000	472.000000	
mean	177.201335	4.004790e+06	7.639388e+10	-207.461419	9.472945	32.827458	8.610633	
std	297.937044	8.178125e+06	1.814206e+11	5487.814466	7.005131	58.121433	22.157837	
min	5.510000	3.300000e+01	4.670000e+09	-118732.400000	-4.400000	2.760000	0.110000	
25%	63.977500	8.881818e+05	1.691250e+10	11.825000	4.125000	13.855000	2.837500	
50%	109.345000	1.675000e+06	3.314500e+10	19.005000	7.685000	22.325000	5.280000	
75%	201.560000	3.770000e+06	6.552750e+10	32.630000	13.475000	32.650000	9.105000	
max	4687.600000	7.365000e+07	2.350000e+12	9183.040000	45.090000	842.120000	447.810000	

Figure 2 Result of the describe() function to retrieve a statistical overview of the data

A pair plot is employed to examine the relationships between numerical features in the dataset. It's noteworthy to pay attention to the line of best fit in this analysis. By drawing a line of best fit concerning other numerical features, this visualization can be categorized as linear. In a linear relationship, if one of the features is high, then the target variable is also high. An important observation from the graph is the presence of 'outliers' on the left side. These outliers might significantly influence the data prediction.

The data distribution of each feature can be further visualized using a box plot (see Figure 3). The box plot confirms that numerous numerical features exhibit a notable presence of outliers. However, the outliers will be kept as in the real world, these metrics also tend to vary a lot from one stock to another.

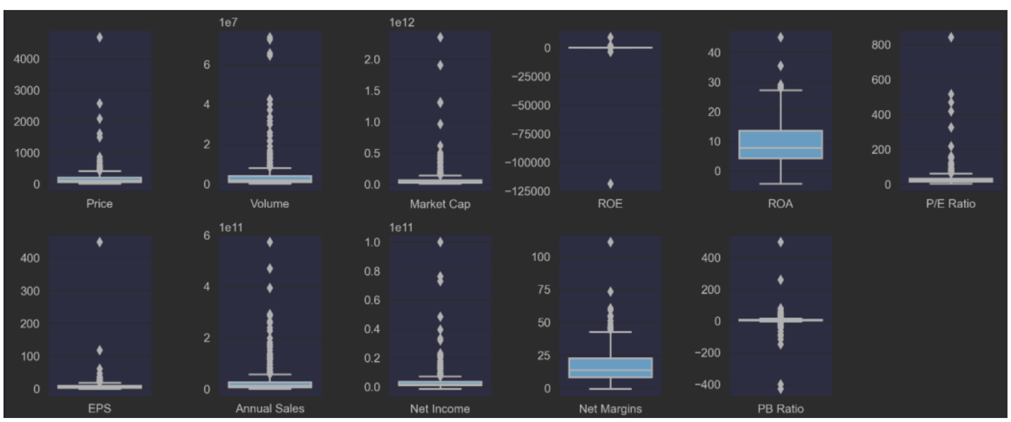

Figure 3 Box plot for the numerical features from the data

After gaining a deeper understanding of the numerical features, the same process will be conducted for the categorical feature, which is the Exchange and Rating feature. The describe(include='object') function is employed to extract statistical information about the categorical features of the dataset. This includes insights into unique values, frequencies, and other relevant statistics.

Subsequently, the distribution of unique values within the categorical features is examined through a bar graph (refer to Figure 4). The bar graph showed that most stocks in the dataset are from the NYSE exchange and most have a rating of "Moderate Buy" with the "Reduce" rating being the rarest. Furthermore, a pie chart is utilized to visually represent the numerical proportions of the data in percentages (see Figure 5).

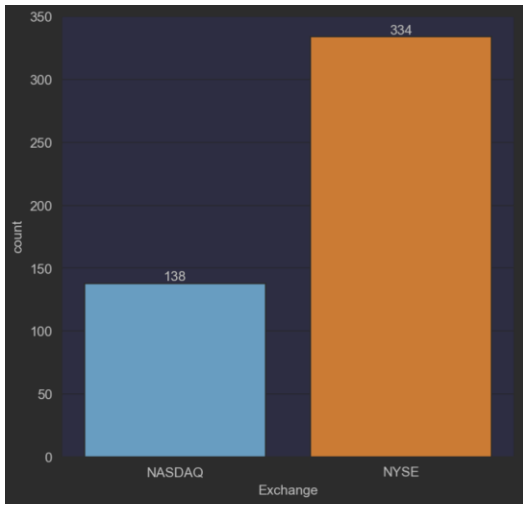

Figure 4 Distribution of the exchange feature using a bar graph

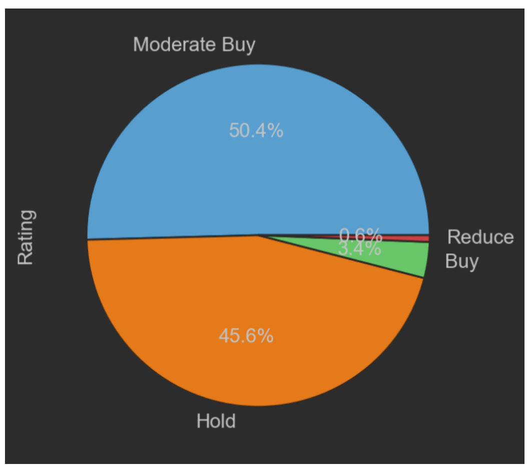

Figure 5 Distribution of the rating feature in percentages using a pie chart

After a good understanding of the data, the data will be pre-processed for training. The purpose of pre-processing is to prepare the data by cleaning and organizing it, making it suitable for constructing and training a machine learning model. Following this, the Rating and Exchange features will be encoded since their values are ordinal and nominal, respectively. The identification and separation of data into dependent and independent variables will then be carried out, and the data will be split into training and testing sets. Additionally, feature scaling will be applied to the independent variables to enhance accuracy.

Both categorical features are still in the form of a string. Thus, encoding will be done to convert these string representations into numerical values during the dataset preprocessing phase. This conversion is essential for machine learning algorithms to effectively interpret and analyse the categorical data. The Rating feature is converted manually, where the “Reduce”, “Hold”, “Moderate Buy”, and “Buy” values are converted into 1, 2, 3, and 4 respectively, while the Exchange feature is encoded using the OneHotEncoder method by calling the get_dummies() function from Pandas. This will remove the Exchange column and create 2 new columns in its place, the Exchange_NASDAQ and the Exchange_NYSE column with the value 0 if it's false and 1 if it's true.

The subsequent phase of data pre-processing involves segregating the features into dependent and independent variables. The target variable, denoted as 'y', corresponds to the stock’s “Rating” which the objective is to classify the rating of the stock. Consequently, the factors influencing the target variable are identified as the independent variables, denoted as 'X'. However, a high number of independent variables can impede the effectiveness of classification models, causing instability in the process of selecting relevant features [13].

To identify the features for training the model, the SelectKBest method from sklearn is employed. This method determines the *k* columns with the highest scores, indicating their significant contribution to the target variable. The target variable is set as the Rating column, while all other columns serve as input features. The score function for SelectKBest is set to *f_classif*, as it supports negative values. The parameter *k* is set to 6, signifying that the method will return the top 6 features based on their scores. The top 6 most significant features returned are the Price, Market Cap, P/E Ratio, Annual Sales, Net Income, and PB Ratio.

After balancing the target variable values, the next step involves feature scaling for the variables. Feature scaling stands out as a crucial data pre-processing step in machine learning to promote easy and fair comparison among values. A study by Ozsahin et al. [14] found that feature scaling improved the performance of their *k*-NN and SVM model. Therefore, a feature scaling technique known as normalization will be applied. The chosen features are subsequently normalized using the MinMaxScaler() function. This step is essential because the dataset contains values with a broad range. For instance, the Annual Sales columns may encompass numbers ranging from millions to billions, while the P/E Ratio column has a narrower range of 2 to 842.

After normalizing the data, it is further split into 4 parts. The features will be split into the X_train and X_test DataFrame while the target will be split into the y_train and y_test DataFrame. The split is done with 0.25 as the test size (25% of the data will be used for testing and 75% of the data will be used for training) and 0 as the random_state value.

Five popular classification methods are chosen. All of the methods chosen will be tested first for parameter tuning to find the best parameter for the model. These methods include *k*-NN (*k*-Nearest Neighbors), Gaussian Naive Bayes, SVM (Support Vector Machine), Decision Tree, and Random Forest. The performance of the resulting models will be compared using the evaluation metrics that will be chosen and explained during the evaluation step so that the best model for classification can be found.

Prior to creating the model, it is important to identify the optimal parameters that can maximize the performance of the machine learning model. To accomplish this task, a library within sklearn known as GridSearch is employed. Each classification algorithm comes with its unique set of parameters, distinguishing them from one another. Hence, when utilizing GridSearch, there is no need to repeatedly optimize the same parameters for every classification algorithm.

Evaluating the performance of the model is a stage to measure the quality of the used algorithm. The evaluation is measured using two metrics: accuracy and weighted F1-score.

The consideration of class imbalance in evaluating model performance is crucial, as highlighted by Luque et al. [13]. While accuracy is a commonly used metric, its effectiveness can be misleading in datasets with imbalanced class distribution [16]. In dealing with imbalanced datasets, the weighted F1-score is particularly valuable, as it assigns greater significance to classes with more instances [15]. Accuracy is selected to gauge the overall correctness of the model in predicting the correct rating. The F1-score is utilized to measure the model's performance by considering both precision and recall, often used for comparing performance across models.

The choice of weighted F1-score over other types of F1-score is motivated by its ability to assign greater importance to classes with more examples in the dataset [17]. This makes it particularly suitable for datasets with imbalances. In this scenario, the 'Reduce' and 'Buy' ratings exhibit significant imbalance when compared to the 'Hold' and 'Moderate Buy' ratings.

Table 1 Results of the evaluation metrics for each algorithm

Classifier Algorithm	Accuracy	Weighted F1-Score
k-Nearest Neighbors (*k*-NN)	0.618644	0.605011
Gaussian Naive Bayes	0.516949	0.474300
Support Vector Machine (SVM)	0.567797	0.554887
Decision Tree	0.525424	0.508889
Random Forest	0.550847	0.540375

The five experiments in Table 1 were carried out in this study, one for each algorithm. The *k*-NN algorithm demonstrated the best performance, achieving an accuracy of 0.618644 and a weighted F1-score of 0.605011. The Gaussian Naïve Bayes algorithm performed worse, achieving an accuracy of 0.516949 and a weighted F1-score of 0.474300. This algorithm also showed a pronounced disparity between the accuracy and the weighted F1-score, indicating that the model may be struggling with minority classes. SVM showed the second-best performance, achieving an accuracy of 0.567797 and a weighted F1-score of 0.554887. Decision Tree only achieved an accuracy of 0.525424 and a weighted F1-score of 0.508889 while Random Forest achieved an accuracy of 0.550847 and a weighted F1-score of 0.540375.

Overall, the evaluation results for *k*-NN, Gaussian Naive Bayes, SVM, Decision Tree, and Random Forest were unfavorable. Among the five models, *k*-NN demonstrated the best performance, boasting the highest accuracy and weighted F1-score. Despite its superiority, all models are deemed to provide subpar predictions, with only achieving the highest accuracy at 0.618644. It must be noted that stock prediction models tend to have lower performance. Sonkavde et al. [18] compared the performance of existing stock prediction algorithms, and most algorithms, such as the ones used in this study, only achieved an F1-Score in the range of 60% to 70%. The overall low performance may be attributed to the fact that there are numerous external factors affecting the performance of a stock. Sundar et al. [19] identifies macroeconomic indicators, company performance, investor sentiment, and external events as key influencers. Mohan et al. [20] and Khan et al. [21] both highlight the importance of external factors such as news sentiment and social media in stock prediction, while Serafeim and Yoon [22] also suggest that ESG (Environmental, Social and Governance) ratings may be useful in this context. These studies collectively suggest that the accuracy of stock prediction may be improved by considering external factors aside from the stock price and the company's internal factors themselves. However, this approach can also introduce complexity, data availability challenges, and a risk of overfitting, which may hinder model interpretability and performance on unseen data.

Conclusions

The *k*-NN model exhibits the highest accuracy and weighted F1-score at 0.619 and 0.605, respectively. Consequently, the *k*-NN model emerges as the optimal choice for predicting stock analyst ratings from the extracted dataset. However, it is noteworthy that despite it achieving the highest accuracy, the overall accuracy remains relatively low at 0.619, signifying the model's ability to accurately predict only 61.9% of the testing data. This limitation may stem from the fact that stock analysts often base their recommendations on factors beyond the stock itself, such as news, sustainability and, ethical concerns for the company. As a recommendation for future research, considering the incorporation of external factors such as news sentiment, social media, and ESG ratings may enhance the predictive capabilities of the model, but the risk of increased complexity, data availability, and overfitting must be considered.

References

[1] A. Kuriakose and P. B. Sajoy, "Digital Transformation in the Stockbroking Industry and its Role in Strong Retail Investor Participation in the Indian Stock Market," *The Management Accountant Journal*, vol. 57, no. 8, p. 50, Aug. 2022, doi: 10.33516/maj.v57i8.50-54p.

[2] Dr. D. Pathak, "Intrinsic Value in Assessing the Fairness of IT Stock Price using Fundamental Analysis", *BSSS Journal of Management*, vol. 12, no. 1, pp. 23–34, Sep. 2021, doi: 10.51767/jm1203.

[3] P. Wnuczak, "Profitability of investment strategies developed on the basis of buy and sell recommendations," *Journal of Economics and Management*, vol. 43, pp. 317–338, 2021, doi: 10.22367/jem.2021.43.15.

[4] N. S. Soraya and H. Hendry, "Komparasi linear regression, random forest regression, dan multilayer perceptron regression untuk prediksi tren musik TikTok," *AITI*, vol. 20, no. 2, pp. 191–205, Aug. 2023, doi: 10.24246/aiti.v20i2.191-205.

[5] D. Kurniadi, Y. H. Agustin, H. I. N. Akbar, and I. Farida, "Penerapan Algoritma k-Means Clustering untuk Pengelompokan Pembangunan Jalan pada Dinas Pekerjaan Umum dan Penataan Ruang," *AITI*, vol. 20, no. 1, pp. 64–77, Mar. 2023, doi: 10.24246/aiti.v20i1.64-77.

[6] T. Wiradinata, F. Graciella, R. Tanamal, Y. S. Soekamto, and T. R. D. Saputri, "Post-Pandemic Analysis of House Price Prediction in Surabaya: A Machine Learning Approach," *Journal of Southwest Jiaotong University*, vol. 57, no. 5, pp. 562–573, Oct. 2022, doi: 10.35741/issn.0258-2724.57.5.45.

[7] X. Ji, J. Wang, and Z. Yan, “A stock price prediction method based on deep learning technology,” *International Journal of Crowd Science*, vol. 5, no. 1, pp. 55–72, Apr. 2021, doi: 10.1108/IJCS-05-2020-0012.

[8] Z. K. Lawal, H. Yassin, and R. Y. Zakari, “Stock Market Prediction using Supervised Machine Learning Techniques: An Overview,” in *2020 IEEE Asia-Pacific Conference on Computer Science and Data Engineering (CSDE)*, IEEE, Dec. 2020, pp. 1–6. doi: 10.1109/CSDE50874.2020.9411609.

[9] S. Kompella and K. C. Chilukuri, “Stock Market Prediction using Machine Learning Methods,” *Inernational Journal of Computer Engineering and Technology*, vol. 10, no. 3, May 2019, doi: 10.34218/IJCET.10.3.2019.003.

[10] M. Misra, A. P. Yadav, and H. Kaur, “Stock Market Prediction using Machine Learning Algorithms: A Classification Study,” in *2018 International Conference on Recent Innovations in Electrical, Electronics & Communication Engineering (ICRIEECE)*, IEEE, Jul. 2018, pp. 2475–2478. doi: 10.1109/ICRIEECE44171.2018.9009178.

[11] G. Attanasio, L. Cagliero, and E. Baralis, “Leveraging the explainability of associative classifiers to support quantitative stock trading,” in *Proceedings of the Sixth International Workshop on Data Science for Macro-Modeling*, New York, NY, USA: ACM, Jun. 2020, pp. 1–6. doi: 10.1145/3401832.3402679.

[12] J. C. Fernández, M. Carbonero, P. A. Gutiérrez, and C. Hervás-Martínez, “Multi-objective evolutionary optimization using the relationship between F1 and accuracy metrics in classification tasks,” *Applied Intelligence*, vol. 49, no. 9, pp. 3447–3463, Sep. 2019, doi: 10.1007/s10489-019-01447-y.

[13] É. Perthame, C. Friguet, and D. Causeur, “Stability of feature selection in classification issues for high-dimensional correlated data,” *Stat Comput*, vol. 26, no. 4, pp. 783–796, Jul. 2016, doi: 10.1007/s11222-015-9569-2.

[14] D. U. Ozsahin, M. Taiwo Mustapha, A. S. Mubarak, Z. Said Ameen, and B. Uzun, “Impact of feature scaling on machine learning models for the diagnosis of diabetes,” in *2022 International Conference on Artificial Intelligence in Everything (AIE)*, IEEE, Aug. 2022, pp. 87–94. doi: 10.1109/AIE57029.2022.00024.

[15] A. Luque, A. Carrasco, A. Martín, and A. de las Heras, “The impact of class imbalance in classification performance metrics based on the binary confusion matrix,” *Pattern Recognit*, vol. 91, pp. 216–231, Jul. 2019, doi: 10.1016/j.patcog.2019.02.023.

[16] J. Novakovic, A. Veljovic, S. Ilić, Ž. M. Papic, and T. Milica, “Evaluation of Classification Models in Machine Learning,” *Theory and Applications of*

Mathematics & Computer Science, vol. 7, pp. 39–46, 2017, [Online]. Available: https://api.semanticscholar.org/CorpusID:125586327

[17] A. K. Gupta, N. Tatbul, R. Marcus, S.-T. Zhou, I. Lee, and J. E. Gottschlich, “A Skew-Sensitive Evaluation Framework for Imbalanced Data Classification,” 2020. [Online]. Available: https://api.semanticscholar.org/CorpusID:222310663

[18] G. Sonkavde, D. S. Dharrao, A. M. Bongale, S. T. Deokate, D. Doreswamy, and S. K. Bhat, “Forecasting Stock Market Prices Using Machine Learning and Deep Learning Models: A Systematic Review, Performance Analysis and Discussion of Implications,” *International Journal of Financial Studies*, vol. 11, no. 3, p. 94, Jul. 2023, doi: 10.3390/ijfs11030094.

[19] S. Sundar, Mr. B. Dhyani, and D. P. Chhajer, “Factors Affecting Stock Market Movements: An Investors Perspective.,” *European Economic Letters*, 2023, [Online]. Available: https://api.semanticscholar.org/CorpusID:258137619

[20] S. Mohan, S. Mullapudi, S. Sammeta, P. Vijayvergia, and D. C. Anastasiu, “Stock Price Prediction Using News Sentiment Analysis,” in *2019 IEEE Fifth International Conference on Big Data Computing Service and Applications (BigDataService)*, IEEE, Apr. 2019, pp. 205–208. doi: 10.1109/BigDataService.2019.00035.

[21] W. Khan, M. A. Ghazanfar, M. A. Azam, A. Karami, K. H. Alyoubi, and A. S. Alfakeeh, “Stock market prediction using machine learning classifiers and social media, news,” *J Ambient Intell Humaniz Comput*, vol. 13, no. 7, pp. 3433–3456, Jul. 2022, doi: 10.1007/s12652-020-01839-w.

[22] G. Serafeim and A. Yoon, “Stock Price Reactions to ESG News: The Role of ESG Ratings and Disagreement,” *SSRN Electronic Journal*, 2021, doi: 10.2139/ssrn.3765217.