Iranian Journal of Finance

Print ISSN 2676-6337
Online ISSN 2676-6345

Comparative Analysis of Machine Learning Algorithms in Predicting Jumps in Stock Closing Price: Case Study of Iran Khodro Using NearMiss and SMOTE Approaches

Ahmad Jafarnejad*

*Corresponding Author, Prof., Prof., Department of Industrial Management, Faculty of Industrial Management and Technology, College of Management, University of Tehran, Tehran, Iran. (Email: afarnjd@ut.ac.ir)

Arman Rezasoltani

Ph.D. Candidate, Department of Industrial Management, Faculty of Industrial Management and Technology, College of Management, University of Tehran, Tehran, Iran. (Email: armanrezasoltani@ut.ac.ir)

Amir Mohammad Khani

Ph.D. Candidate, Department of Industrial Management, Faculty of Industrial Management and Technology, College of Management, University of Tehran, Tehran, Iran. (Email: amir.mo.khani@ut.ac.ir)

Iranian Journal of Finance, 2025, Vol. 9, No.3, pp. 27-54.
Publisher: Iran Finance Association
doi: https://doi.org/10.30699/ijf.2025.491324.1496
Article Type: Original Article
© Copyright: Author(s)
Type of License: Creative Commons License (CC-BY 4.0)

Received: July 12, 2024
Received in revised form: December 26, 2024
Accepted: March 24, 2025
Published online: July 20, 2025

Abstract

Predicting stock price fluctuations has always been one of the most important financial challenges due to the complexities of financial data and nonlinear market behavior. This research aimed to analyze and compare the performance of machine learning algorithms in predicting the closing price jump of Iran Khodro Company shares. Two different methods of managing unbalanced data, NearMiss and SMOTE, were used to overcome the challenge of unbalanced data. The results showed that the NearMiss method outperformed SMOTE by balancing precision and recall in machine learning models. The CatBoost model was recognized as the best machine learning model in this study due to its stable performance in NearMiss and SMOTE methods. The CatBoost model showed a perfect balance between evaluation indicators in the NearMiss method, with an accuracy of 91.46% and an F1 score of 91.29%. This model also had high precision (93.18%) and acceptable recall (89.52%), which showed the ability to detect jumps and avoid wrong predictions correctly. On the other hand, in the SMOTE method, the Random Forest model was superior, with an accuracy of 85.08%. These results show that a combination of unbalanced data management methods and advanced machine learning algorithms can significantly improve the accuracy of price volatility prediction. The results of this research can help investors and financial analysts make better decisions in risk management and optimizing investment strategies.

Keywords: Machine Learning, Imbalanced Data Handling, NearMiss, SMOTE, Stock Price Prediction

JEL Classification: C53, G17, C45, C63, C81

Introduction

One of the foremost and most significant challenges in the finance and investment fields has always been the analysis of stock selections. Stock markets are some of the most complex arenas for analysis, characterized by high volatility and nonlinearity, depending on factors such as economic conditions, news, and public sentiment (Johnson et al., 2003; Pfluger et al., 2020). Investors seek new tools that give them the most accurate stock price forecasts to make more efficient investment and risk management decisions. Meanwhile, technological progress in machine learning and artificial intelligence has brought forward more accurate stock price prediction by advanced tools (Methan Prasad & Gunasekaran, 2020). Some algorithms are highly effective at predicting stock prices as they can handle much data and find hidden patterns (Presar et al., 2023; Gu et al., 2020). This includes methods such as Recurrent Neural Networks (RNN), Long Short-Term

Memory Models (LSTM), Support Vector Machines (SVM), and hybrid methods such as SMOTE (Synthetic Minority Oversampling Technique) and NearMiss to cope with the challenges of imbalanced data analysis. Additionally, these tools have expanded the scope of analysis of complex financial data and enhanced the accuracy of financial forecasts (Gupta & Ahmed, 2019). The first key challenges are market volatility, high data volume, or imbalanced training datasets.

It has been shown in previous research reviews that most research only covers balanced data or simple data analysis without showing good performance for imbalanced data feature cases of unexpected fluctuation. Additionally, many of the models examined are targeted at a single algorithm, and few such models have conducted systematic comparisons or assessments of combined approaches. Some studies independently used traditional data augmentation methods like SMOTE and NearMiss. A limitation of this is that no comprehensive study has been undertaken assessing the impact of these methods in conjunction with machine learning algorithms on imbalanced data (Prasad et al., 2021). Moreover, a lack of research on Iran's financial market and company shares, such as Iran Khodro, leaves a significant research gap. This research paper aims to investigate and evaluate the performance of various machine learning algorithms in handling stock price fluctuations, focusing on the closing stock price of Iran Khodro Company. We explore the performance of machine learning models enhanced via imbalanced data methods (such as SMOTE and NearMiss). This study aims to bridge gaps from previous research, develop new insights into stock market analysis, and offer new, more precise decision-making tools. Theoretically and practically, this research is fundamental. Theoretically, this research bridges existing literature gaps in imbalanced data analysis and hybrid methods in machine learning. This research may develop more accurate analytical tools that analysts and investors can use to make correct decisions in the dynamic stock market (Bhamar et al., 2023).

Unlike previous studies focusing on price forecasting, we present a new approach for predicting stock price jumps in the Iranian financial market. Due to the ±5% daily price fluctuation and the resulting data imbalance, stock price movements are quite different from other markets. We systematically evaluate the impact of SMOTE and NearMiss in addressing this challenge, ensuring that synthetic data generation aligns with real market behavior. Additionally, we explore the implications of these resampling techniques for financial modeling under liquidity constraints by comparing the results achieved using 3D PCA visualization. This helps us determine the appropriateness of these techniques

for the feature space. In addition, feature engineering techniques are used to create meaningful predictive variables, consisting of the previous day's closing price, short-term price change, moving averages, volatility, and trading volume, to improve the model's ability to identify stock price jumps. We also explore the importance of features with Random Forest and CatBoost and identify the most important factors in driving stock price movements. These insights contribute to a more interpretable and practical predictive framework, bridging the gap between machine learning techniques and financial market behavior. The second novel aspect of this study is the focus on actual data from the Iranian stock market. This research paper is finally organized into sections, starting with the introduction, which networked the research background, objectives, and justifications for its undertaking. The literature review is in section two, and another section covers related works. The data, models, and analytical methods are described in the methodology section. Empirical findings and statistical analysis are presented in the results section. Conclusions and recommendations for future research are presented in the final section. This paper is structured to analyze the research topic thoroughly and provides clear guidelines for conducting similar investigations in this regard.

Literature Review

A considerable number of studies have been conducted on stock price forecasting, as well as the use of machine learning and deep learning algorithms. Utilizing varying methods and datasets, these studies have sought to address the challenges of stock market fluctuation analysis and prediction, striving to improve forecasting precision. Table 1 presents a systematic review of the primary studies conducted in this field. This review examines the objectives, models used, datasets, and main results of these studies, providing a foundation for identifying research gaps and proposing new directions for future research.

Table 1. Research background

Authors	Title of the article	Objectives	Model used	Dataset	Conclusion
Gupta and Ahmad (2019)	Predicting stock price trends using long short-term memory (LSTM) networks	Predicting stock price trends using LSTM	LSTM	Historical stock price data	The LSTM model has shown high accuracy in predicting stock trends.
Prasad et al. (2021)	Stock Price Forecasting	Comparing different	Kalman Filters‘	Time series	The combined Kalman-

	Using Statistical Models and Machine Learning: A Comparative Analysis	models for predicting stock prices	XGBoost ، ARIMA	financial data	XGBoost model showed the best performance.
Khairi et al. (2019)	Stock price forecasting using a combined technical, fundamental, and new approach	Predicting stock price fluctuations by combining technical, fundamental, and news data	J48 (Decision Tree) ، Bagging	Technical and fundamental stock data	The combination of technical and news data showed high accuracy in forecasting.
Mehta et al. (2021)	Applying social media sentiment analysis to improve stock market forecasting with deep learning.	Social media sentiment analysis for stock prediction	SVM ، LSTM ، Naive Bayes	Social media data and news	The use of sentiment analysis improved the prediction accuracy.
Heydari and Amiri (2022)	Examining the power of artificial intelligence-based models in predicting stock price trends on the Tehran Stock Exchange	Evaluating the accuracy of machine learning models in predicting stock price trends	Neural networks, logistic regression, K-nearest neighbor, support vector machine	Data of the 150 largest companies on the Tehran Stock Exchange (2011-2019)	Deep learning models perform better than others, with an accuracy of around 70 to 80 percent in predicting short-term stock price trends.
Shariffar et al. (2022)	Application of deep learning architectures in stock price prediction (Convolutional Neural Network (CNN) approach)	Investigating the ability of different CNN algorithm architectures to predict stock prices	Convolutional neural network	Isfahan Zob Ahan Company daily stock data	Using CNN with a max pooling layer has a MAPE error of 1.79% and NRMSE of 2.71%, indicating its better performance than other architectures and the RNN algorithm.
Mathanprasad and Gunasekaran (2022)	Stock market trend analysis and market forecast performance evaluation with a machine learning approach	Predicting stock market price fluctuations using machine learning	Machine learning classification	Real-time stock market data	Price prediction accuracy has improved to 94.17%.

Subekti and Saepudin (2022)	Cross-sectional machine learning approach to predict stock returns of LQ45 index	Predicting stock returns using cross-sectional machine learning	Cross-sectional ML	Indonesia LQ45 Stock Data	The cross-sectional model performed better than the time series methods.
Khandagale et al. (2023)	Stock price prediction with machine learning using comparative analysis of the Random Forest algorithm	Comparative analysis of machine learning algorithms for price prediction	Random Forest, Linear Regression, Decision Tree	Historical Stock and ETF Data	The Random Forest algorithm has shown the best prediction performance.
Parashar et al. (2023).	Machine learning framework for stock prediction using sentiment analysis.	Using sentiment analysis to predict stock prices	Random Forest ، Multinomial Naive Bayes	Financial news headline data	Sentiment analysis improved prediction accuracy.
Bhamare et al. (2023)	Predicting stock market closing prices using deep learning and machine learning algorithms	Predicting closing stock prices	SVM ، Random Forest ، LSTM	HDFC, AXIS, ICICI Bank Stock Data	The Random Forest model showed higher accuracy than other models.
Izsák et al. (2023).	Evaluation of stock price prediction based on support vector machine (SVM)	Evaluating stock price prediction models with SVM	SVM	High-volume stock data	The SVM algorithm showed high performance in analysis and prediction.
Gholami and Shams Qarneh (2024)	Presenting a model for stock price prediction based on optimized CNN-LSTM in the Tehran Stock Exchange	Presenting a hybrid CNN-LSTM model for stock price prediction	(CNN) ، (LSTM)	Data for 10 stocks from the Tehran Stock Exchange (2013-2023)	The proposed LSTM-CNN model with hyperparameter optimization using the PSO algorithm performs better than other models.
Rezaian et al. (2024)	Developing a comprehensive model for predicting stock prices in the stock market using an interpretive structural modeling approach	Identifying the metrics that affect stock price prediction and developing a comprehensive model	Interpretive Structural Modeling (ISM)	Tehran Stock Exchange data	Identifying 54 stock price prediction criteria and providing a comprehensive model for prediction

Background of research consists in saying that machine learning and deep learning models and algorithms have been widely used to forecast stock prices, and there have been many previous studies. A variety of algorithms, from neural networks, support vector machines, random forests, convolutional neural networks, long short-term memory models, etc, were employed in these studies. Several aspects of such studies were reviewed and were successful, including the large variety of algorithms and approaches used. Factors like MAPE, NRMSE, and forecast accuracy in many studies can be used to compare the model's performance accurately. Moreover, such studies also considered various data types, including historical, technical, fundamental, news, and sentiment analysis, to account for various factors that determine the forecasting model for the stock price.

Furthermore, Gholami and Shams Qarneh (2024) and Prasad et al. (2021) conducted work with hybrid models that combine different algorithms to achieve a better result. Nevertheless, there are significant weaknesses in the research background. The problem of imbalanced data is not adequately addressed, in which case biased outcomes may occur, as well as reduced accuracy of forecasting models. Furthermore, most previous research has been associated with general stock market data since only a few studies have analyzed the sufficiency of the stocks of specific companies or industries. As a result, the outcomes are generalized (resulting in less practical value). While Shariffar et al. (2022) used one of the specific models like CNN, they did not form a thorough comparison between the models and other algorithms. In contrast, other studies did not investigate data-driven approaches such as SMOTE and NearMiss to help improve the model's accuracy in case of imbalanced data, which has significant potential as a method. However, limited attention has been paid to other challenges — model stability and generalizability to changing market conditions.

Several gaps in the existing literature are addressed in the present study. The present study begins by addressing something that was only vaguely considered by the overwhelming majority of prior works: the need to handle the cases of imbalanced data. The second gap in the literature is that studies have focused mainly on generalized market data, which offer far less support and no attention to node-specific analyses of specific company stocks. In this vein, this study fills this gap by studying the stock of Iran Khodro Company. Third, while some studies propose hybrid models, there is still insufficient research on combining machine learning algorithms and data preprocessing techniques. This level of integration can provide a meaningful step toward more accurate forecasting.

Furthermore, most studies have paid more attention to forecasting trends or overall stock price predictions rather than analyzing the fluctuations in closing stock prices. This study purports to contribute substantially to science and practice by focusing attention on a single item. Finally, this study takes a broader comparative approach, evaluating the performance of several machine learning algorithms in a real-world scenario, where previous studies were largely lacking in complete comparisons with different algorithms. A novel aspect of this research is that it addresses a gap in the literature by running a study on NearMiss and SMOTE applied on Iran Khodro stocks and a comprehensive comparison of multiple different machine learning algorithms. This seminal work helps to further knowledge in stock price prediction research. This research could pave the way for developing more efficient models for analyzing Iranian stock market data to provide practical solutions to the problem of imbalanced data and to improve forecasts.

Research Methodology

This research study compares various machine learning algorithms to determine their capability to predict the price jumps of Iran Khodro Company (Khodro) shares from the time of their entry into the stock market up to September 21, 2022. This research attempts to study the validity of such algorithms in predicting stock price jumps from historical stock data, employing different machine learning algorithms. Two data balancing approaches were adopted to address challenges presented by the dataset, such as class imbalance: SMOTE and NearMiss. The structure of the research is presented here:

- Data: The dataset was gathered from Iran Khodro Company shares' trading history on the stock exchange, from the first day of the listing to September 21, 2022. For each trading day, the closing price, trading volume, trading value, number of transactions, opening price, highest price, and lowest price are recorded, making up seven features in the dataset.
- Data Preprocessing: Preprocessing the collected data so it is of quality and consistency. In this process, duplicate records were removed, missing values were handled, and the data was normalized in machine learning models (Chandar, 2024).
- Feature Engineering: A critical step involved in predicting a stock price jump is feature engineering. This study extracts and calculates several features from historical stock data to leverage it more accurately for predictions of price jumps without relying on future data. Specifically, the chosen features represent instantaneous fluctuations, general trends, and

historical price behaviors as input for machine learning models (Shen & Shafiq, 2020). The range of the stock price fluctuations in the Iranian Stock Exchange, usually within a range of ±5%, transforms price jumps into those price changes that are more than ±4% above and beyond the closing price. It allows for detecting significant price changes and discards the most minor 'day-to-day' changes. According to this definition, the following features were created to predict price jumps based on this definition.

1. Short-term closing price changes

This feature aims to capture short-term price fluctuations – the percentage change in the closing price over one or two days preceding. It seeks to uncover downward spikes and short-term trends. Including this feature allows the model to distinguish better patterns related to price jumps and momentary price variations. The feature is computed using the following equation:

$$Short-term\ closing\ price\ changes = \frac{Closing\ price\ of\ the\ previous\ day - Closing\ price\ of\ two\ days\ prior}{closing\ price\ of\ two\ days\ prior} \quad (1)$$

2. Short-term moving average

This feature calculates the average closing price over the last five days, capturing overall price trends. Averaging out this noise from daily fluctuations helps you identify broader market patterns. This feature allows the model to represent historical data compactly and distinguish between bullish, bearish, and neutral market trends. The following formula is used to compute the average:

$$Short-term\ moving\ average = \frac{\sum_{i=1}^{5} Closing\ price\ of\ i\ days\ prior}{5} \quad (2)$$

3. Closing price of the day prior

This directly provides the previous day's closing price to the model, allowing it to work out the relationship between the prior day's value and the current day's price. It is used as a reference point for detecting short-term changes.

4. Price volatility of the day prior

This feature calculates the price volatility of the previous day by measuring the difference between the highest and lowest prices for that day. Its purpose is to assess the intensity of volatility from the previous day. High volatility often indicates emotional behavior or unusual market movements, which may be associated with price spikes.

5. Trading volume of the day prior

This feature represents the previous day's trading volume and is designed to identify the relationship between trading volume and price changes. High trading volume often reflects increased investor activity, which may signal potential price changes.

This combination of features offers comprehensive information for identifying patterns associated with price jumps and plays a crucial role in enhancing the performance of machine learning models in forecasting.

Feature Importance

Feature importance analysis is pivotal to understanding what variables contribute to predicting a model. To find the most important features for predicting Iran Khodro stock price jumps, We used Random Forest and Catboost and optimized them using Optuna. The engineered features were the previous day's closing price, short-term price change, short-term moving average, previous day's volatility, and previous day's trading volume. In addition, Random Forest and CatBoost performed well on imbalanced data, and both are suitable for real-world financial applications where class distribution is highly skewed. In particular, Random Forest maintained model stability through its ensemble learning while CatBoost, which is what learns from its robust boosting mechanism, tended to deal with the class imbalance effectively. Finally, the optimized hyperparameters for these models improved their predictive performance and generalizability even further. The optimal hyperparameters for Random Forest were found to be n_estimator = 207, max_depth = 14, min_samples_split = 6, min_sample_leaf = 3, max_features = 'sqrt'. For CatBoost, the best config was iterations = 50, depth = 3, learning_rate = 0.021, and l2_leaf_reg = 8.92. With these optimized settings, the models achieved a balanced tradeoff between bias and variance, thus making the models more accurate. Knowing how feature importance affects predictions allows us to make better model selections, highlighting the value of financial forecasting. Through this knowledge, investors and analysts can further sharpen their trading strategies and improve their decision-making in a dynamic, volatile market environment. Figures 1 and 2 show the feature importance rankings of the Random Forest and CatBoost models, respectively.

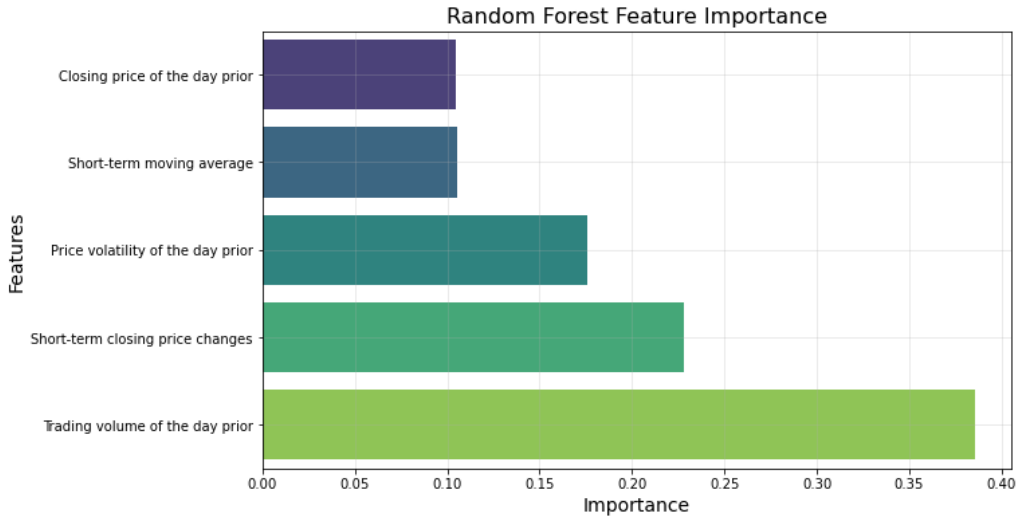

Figure 1. Feature Importance (Random Forest)

The results of the importance of the Random Forest model feature are shown in Figure 1. Results showed that the previous day's trading volume was the most influential predictor and that short-term price change and the previous day's volatility, respectively, followed. Recent price fluctuations and market activity are predominant in forecasting stock price jumps. In contrast, the short-term moving average and the previous day's closing price were not as prominent, suggesting that sequence-based indicators may not be as important for predicting short-term jumps.

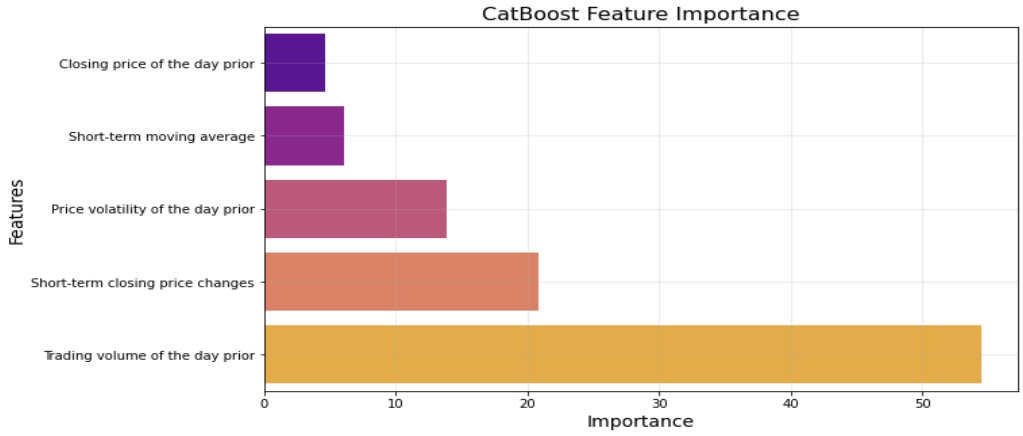

Figure 2. Feature Importance (CatBoost)

In Fig. 2, we can see the importance of the feature of the CatBoost model. As with the Random Forest predictions, trading volume remains the most important feature, indicating that liquidity and trading behavior significantly impact the price. Short-term price change and volatility also exhibit strong importance, consistent with the Random Forest findings. These findings support the notion that short-term price fluctuations, volatility, and trading volume are the principal drivers of stock price changes. The alignment of the two models further supports the robustness of these insights and demonstrates that trading activity is a good predictor in financial markets.

Machine Learning Models

In this study, Machine Learning algorithms have been run to predict price jumps in the stocks of Iran Khodro Company. Both linear and

We used two balancing methods (SMOTE and NearMiss) and two different nonlinear modeling approaches to balance the data and evaluate their ability to handle imbalanced data, respectively. Also, this study employed the Grid Search hyperparameter optimization process to determine the optimal hyperparameter values for each algorithm. The algorithms we used in this research are shown in Table 2.

Table 2. Algorithms used in the research

Algorithm	Brief description	Features and Benefits	Resources
Random Forest	A collection of decision trees that makes predictions by combining their results.	Resistant to overfitting, suitable for complex data	(Barnada et al., 2024; Cosenza et al., 2024)
Gradient Boosting	Models that are built sequentially, with each model reducing the error of the previous model.	Reducing errors from previous models is suitable for accurate prediction.	(Emami and Martinez-Munoz, 2023; Moerman et al., 2018)
XGBoost	A more optimized version of Gradient Boosting with faster speed and additional features such as overfitting prevention.	High speed, ability to fine-tune hyperparameters.	(Liu et al., 2024; Li et al., 2023;Jafarnejad Chaghoshi et al., 2024)
LightGBM	A faster version of XGBoost that uses advanced techniques to increase speed and reduce memory consumption.	Suitable for big data, high performance in classification.	(Bentzak et al., 2020; Hancock and Khoshgafar, 2021)
CatBoost	The gradient Boosting algorithm was optimized for	Reduced preprocessing,	(Dorogosh et al., 2018; Lu and Hu, 2023)

	categorical and numerical data and developed by Yandex.	suitable for categorical data.	
AdaBoost	Models that give more weight to examples with higher errors so that new models focus on them.	Suitable for unbalanced data.	(Tenha et al., 2020)
Logistic Regression	A linear model that predicts the probability of a sample belonging to a particular class.	Simple, fast, and suitable for linear data.	(Kleinbaum and Klein, 2010; James et al., 2021)
KNN	An algorithm that determines the class of a sample based on its proximity to its neighbors.	Nonparametric, suitable for small data.	(Siriopoulos et al., 2023; Zhang et al., 2022)
Decision Tree	An algorithm that classifies data using simple decision rules.	Interpretable, suitable for nonlinear data.	(Kotsiantis, 2011; Costa and Pedreira, 2022)
Naive Bayes	A probabilistic model that works based on Bayes' theorem and assumes independence of features.	Fast, suitable for categorical data.	(Wickramasinghe and Kalutharaj, 2020; Blancocoro et al., 2021)

Data balancing using NearMiss and SMOTE

First, it is inherently difficult to predict stock price jumps accurately since stock prices are notoriously volatile, and too short a sample can lead to invalid predictions. This study utilized two datasets characterized by days with large jumps in price and days with little price change. Our training dataset consists of 4041 non-price-jump days and 439 price-jump days. This imbalance can disrupt the performance of machine learning algorithms, resulting in a bias towards predicting the majority class or days without jumps. Two methods of data balancing, NearMiss, and SMOTE, were deployed to address this challenge.

Undersampling technique NearMiss eliminates redundant samples of the majority class to balance classes. The majority of class samples are selected only when they are closest to the minority class using this method (Wickramasinghe & Kalutharaj, 2020; Blancocoro et al., 2021).

Steps:

1. For each minority class instance, a set of nearest neighbors from the majority class is identified.

2. Retention of the majority of class samples that are most relevant for class

separation.

3. We discard the remaining majority class samples to balance the classes.

Sample augmentation methods like SMOTE can generate new samples for the minority class. Synthetic samples are created by interpolating between existing minority class samples to generate new data points (Al-Reedi et al., 2023).

Steps:

1. Here, we choose k-nearest neighbors for every occurrence of the minority class.

2. The synthetic instances are generated from a linear combination of real instances and their nearest neighbors.

3. Existing instances are augmented with synthetic instances until the classes are proportionally equivalent.

We applied Principal Component Analysis (PCA) in 3 dimensions to further analyze the distribution of the dataset and to visualize the impact of different data balancing techniques. PCA is a dimensionality reduction technique that maps high dimensional data into a lower dimensional space while maximizing the variance. It provides enhanced visualization of data patterns before and after resampling.

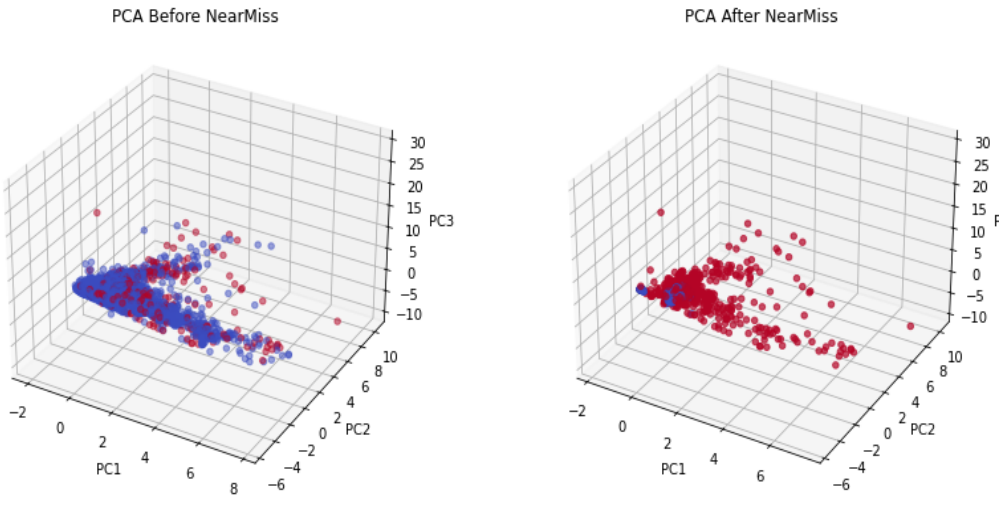

Figure 3. PCA visualization before and after applying NearMiss

Figure 3 PCA projections of the dataset before and after the application of NearMiss. NearMiss is an under-sampling technique that removes samples of the majority class by choosing the instances nearest the minority class. Before we apply NearMiss, the dataset, as seen in the left plot, is highly imbalanced, with the dominant presence of majority class instances. The class distribution is more balanced after applying the NearMiss (right plot), with an apparent decrease in the density of samples of the majority class. This data transformation ensures that the model is not biased towards the most dominant class, enhancing the model's ability to generalize.

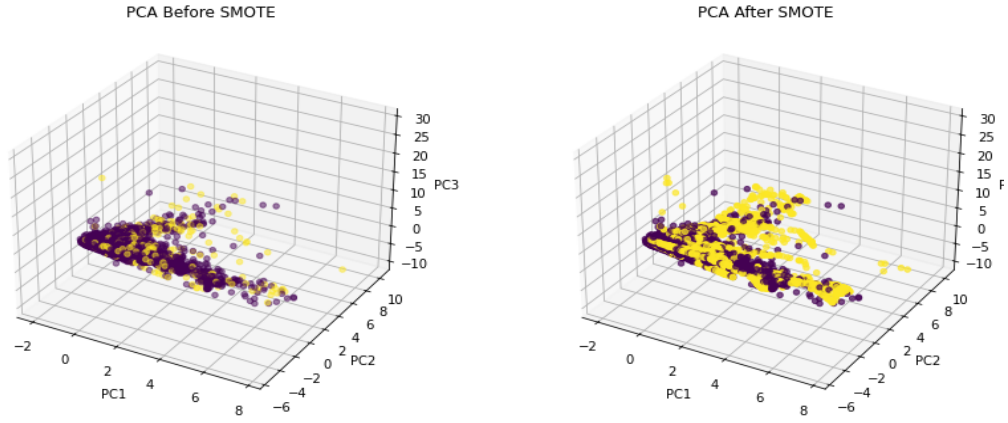

Figure 4. PCA visualization before and after applying SMOTE

Figure 4 PCA visualization before and after applying SMOTE. SMOTE over-sampling technique generates synthetic samples for the minority class using feature space similarities. The left plot shows that the minority class is severely underrepresented before applying SMOTE, which can negatively impact model performance. The synthetic samples, generated using SMOTE (right plot), spread the dataset more evenly, making the data more suitable for training machine learning models. Four primary indicators are used to evaluate the performance of machine learning models in predicting stock closing price jumps. The concepts of True Positive (TP) and True Negative (TN) are used to evaluate these models, representing the model's correct predictions. For example, TP represents the number of correctly identified price jumps, and TN represents the correctly identified days without price jumps (Shen & Shafiq, 2020). On the contrary, False Positive (FP) refers to the number of incorrect predictions where the model predicted a price jump when there was not. False Negative (FN) indicates the number of incorrect predictions where the model fails to predict a price jump when it happened. Performance indicators of

accuracy, precision, recall, and F1 score (Sournavali et al., 2022) are calculated based on these metrics. The machine learning model evaluation indicators are presented in Table 3.

Table 3. Machine learning model evaluation indicators

index	definition	Formula
Accuracy	The ratio of correct predictions (both positive and negative) to the total samples.	$\frac{TP+TN}{TP+FP+FN+TN}$
Precision	The ratio of correct predictions for a class (e.g., price jump) to the total number of samples predicted as that class.	$\frac{TP}{TP+FP}$
Recall	The ratio of correct predictions for a class (e.g., price jump) to the total number of actual examples of that class.	$\frac{TP}{TP+FN}$
F1 score	Harmonized average between Precision and Recall to create a balance between them.	$\frac{2*Precision*Recall}{Precision+Recall}$

The K-Fold (K=5) cross-validation method was used to examine the models' generalizability and ensure they performed stably with different types of data (Arlow & Salis, 2010). This method works as follows:

1. We split the data into five equal parts.

2. One of these parts is used as test data, and the rest as training data at each stage.

3. The process is repeated 5 times, using each part exactly once as test data.

4. The final result of the model is reported as the average of the evaluation indices across all iterations.

Results

In this section, we analyze the results of various machine learning models utilizing two data balancing approaches: SMOTE and NearMiss. The Python programming language was employed in this research, with all models executed on a system equipped with an Intel Core i5-7200U processor, 8 GB of RAM, and Python version 3.12. In Table 4, we present the performance results of the models after applying the NearMiss method.

Table 4. Performance results of models with the NearMiss method

Model	Accuracy	Precision	Recall	F1-Score
Random Forest	0.9077	0.9225	0.8906	0.906
Gradient Boosting	0.9237	0.9428	0.902	0.9219
AdaBoost	0.9089	0.9245	0.8906	0.907
Decision Tree	0.8861	0.8866	0.8861	0.8861
Logistic Regression	0.8884	0.9399	0.8314	0.882
KNN	0.9078	0.9592	0.8519	0.9021
Naive Bayes	0.8713	0.974	0.763	0.8555
XGBoost	0.8895	0.9064	0.8701	0.8874
LightGBM	0.8998	0.9125	0.8861	0.8986
CatBoost	0.9146	0.9318	0.8952	0.9129

The NearMiss results for the CatBoost and Gradient Boosting models showed they performed the best. The best balance among the evaluation metrics was achieved by the CatBoost model, with an accuracy of 91.46% and an F1-score of 91.29%. In addition, the method demonstrated high precision (93.18%) and acceptable recall (89.52%) to identify jumps while minimizing false predictions accurately. The Gradient Boosting model worked nearly as well, achieving the highest accuracy of all the models at 92.37% (with an F1 of 92.19%). The accuracy and recall associated with this performance show that a strong balance exists. The performance of the Random Forest and AdaBoost models was also quite strong, with F1 scores in the range of 90.7%. These models provide high precision and recall, enabling them to be used for jump prediction with high confidence.

On the other hand, we found that the K-Nearest Neighbors (KNN) model yielded the highest precision (95.92%) but exhibited low recall (85.19%), implying that some jumps might have been missed. Due to its ability to avoid false positives, this model is more useful for situations where it is imperative to circumvent false positives. Logistic Regression vs Naive Bayes behaved differently. Logistic regression had 93.99% precision and 83.14% recall, which was highly suitable for applications where accurately predicting positive samples (jumps) is more critical than detecting all instances. In contrast, naive Bayes achieved nearly perfect accuracy (97.4%) but low recall (76.3%), making fewer correct predictions while minimizing false predictions. The Decision Tree model had an F1-score of 88.61%, performing equally well across all metrics but less well than more complex models. Results were preliminary: F1 scores were close to 89% for XGBoost and LightGBM, while CatBoost and Gradient Boosting were slightly better.

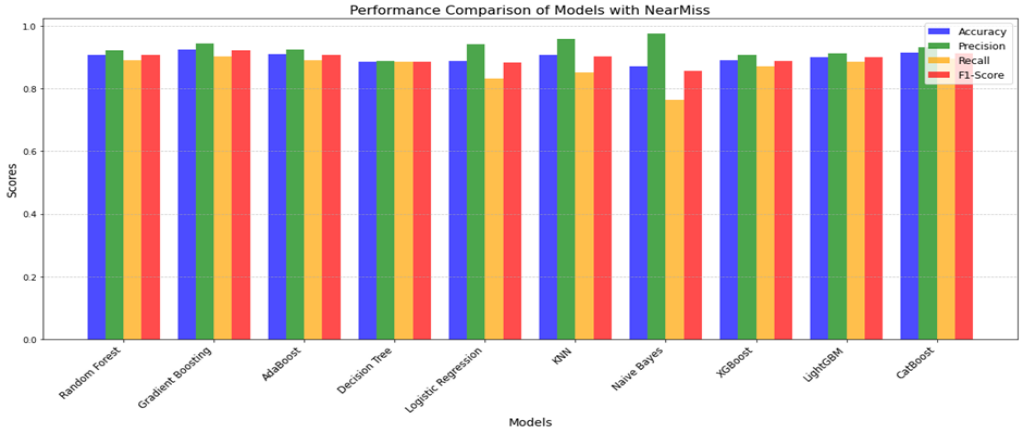

Figure 5. Comparison of models performance with NearMiss

Overall, CatBoost and Gradient Boosting emerged as the best models for predicting price jumps, offering an excellent balance between Precision, Recall, and F1-score. The choice of an optimal model largely depends on the primary objective of the analysis. For instance, if minimizing false predictions (high precision) is the priority, models such as KNN and Naive Bayes are commendable. Conversely, if detecting all price jumps (high recall) is more critical, Gradient Boosting and CatBoost are the preferred models.

Table 5 presents the performance results of models using the SMOTE method.

Table 5. Performance results of models with the SMOTE method

Model	Accuracy	Precision	Recall	F1-Score
Random Forest	0/8508	0/8247	0/8911	0/8566
Gradient Boosting	0/7749	0/7608	0/8023	0/7809
AdaBoost	0/7403	0/7317	0/7587	0/7449
Decision Tree	0/8108	0/7987	0/8317	0/8147
Logistic Regression	0/5613	0/5426	0/9592	0/6889
KNN	0/743	0/737	0/7555	0/7461
Naive Bayes	0/5815	0/8054	0/215	0/3392
XGBoost	0/8372	0/8155	0/8716	0/8426
LightGBM	0/8237	0/7997	0/8636	0/8304
CatBoost	0/8252	0/8017	0/8641	0/8317

In the testing by SMOTE, the Random Forest, CatBoost, and XGBoost models had the highest classification performance. The balance for Random Forest was excellent, with 85.08% accuracy, 85.66% F1-score, 82.47% precision, and 89.11% recall. This balance shows that the model correctly

classifies price jumps and makes fewer incorrect predictions. The performance of CatBoost was also similar to Random Forest, achieving an accuracy of 82.52% and an F1-score of 83.17%. Out of all models tested, CatBoost performed the most successfully at identifying price jumps (recall of 86.41%), though with a precision of 80.17, indicating higher instances of false positives. Among all the models we tested, XGBoost demonstrated the most promising performance regarding 83.72% accuracy and 84.26% F1 score in detecting price jumps. This model's tradeoff lies in accurate detection and error prevention, with an 87.16% recall and robust overall performance.

Our LightGBM model delivered a strong performance, achieving an accuracy of 82.37% and an F1-score of 83.04%. Furthermore, it ranked as one of the most reliable models, with a high recall of 86.36% and a precision of 79.97%. The Gradient Boosting model achieved an accuracy score of 77.49% and an F1 score of 78.09%. Unfortunately, it was not as strong as some best-performing models, like CatBoost and XGBoost. More complex models performed better than simpler models like Decision Tree or KNN. The Decision Tree model managed to maintain a balance, with 81.08% accuracy and 81.47% F1 score. KNN performed relatively poorly, achieving 74.3% accuracy and 74.61% F1-score, but was less able than other models to detect jumps. Logistic Regression and Naive Bayes were found to be the weaker performers. Though Logistic Regression achieved a recall of 95.92%, it had a low accuracy of 56.13%. This imbalance led to many false optimistic predictions with a precision of 54.26%. Naive Bayes performed poorly, with an accuracy of 58.15% and an F1-score of 33.92%, with its recall being just 21.5%.

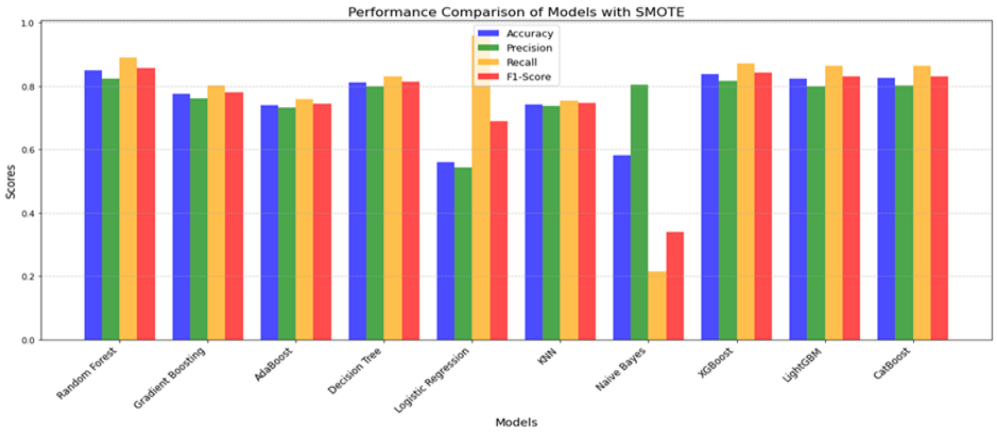

Figure 6. Comparison of model performance with SMOTE

The best-performing models for predicting price jumps are the Random Forest, XGBoost, and CatBoost models, which, except XGBoost, have a fair balance across all evaluation metrics. These models achieve high accuracy and effectively isolate jumps while minimizing false predictions. Models like XGBoost and LightGBM are recommended if maximizing recall (high detection of jumps) is the main objective. On the other hand, if lowering false predictions (high precision) is the focus, Random Forest and CatBoost are the way to go. Logistic Regression and Naive Bayes cannot solve the jump prediction problem because their performance metrics do not behave well with our dataset; therefore, our overall performance is not good. By combining data balancing techniques like SMOTE with robust machine learning models, such as SVM, this analysis demonstrates that it is possible to accurately and reliably detect price jumps.

The NearMiss method is preferred when the primary objective is to minimize false predictions (maximize accuracy). Specifically in the CatBoost and Gradient Boosting models, this method showed stunning accuracy. However, the SMOTE method would be a better option if the goal is to find the number of price jumps at their maximum (maximum recall). SMOTE improved models' recall significantly by generating synthetic samples from the minority class. Overall, the NearMiss method performs better for this study, achieving a better tradeoff between accuracy and recall. The superior models also performed better under this method. This study identified the best machine learning model to use as CatBoost, which performed consistently across both NearMiss and SMOTE methods. The NearMiss and SMOTE models deliver an F1-score higher than 91% for predicting price jumps, exceeding this value significantly. Gradient Boosting and Random Forest also showed strong promise as alternatives. Using NearMiss, Gradient Boosting performed exceptionally, and Random Forest achieved notable results when using SMOTE

Conclusion

Liquidity creation constitutes one of the most crucial subjects in economics and banking literature and influences various factors in banking and macroeconomics. The general framework of the liquidity creation process is enforced through the monetary policy adopted by the central bank. Although this process paves the way to finance projects and meet the liquidity demands of fund applicants, it could expose the bank to instability and failure risk. Furthermore, due to the close interbank connections, this threatening flow can

quickly spread to other banks and cause several problems. Besides, from an economic point of view, the excessive creation of liquidity can initiate the formation of a bubble in asset prices.

However, bank capital and the monetary policy adopted by the central bank could provide a backbone to support banks in managing the abovementioned risk caused by liquidity creation. Therefore, to provide comprehensive insight into the dimensions and impacts of these factors, this study investigated banks admitted to the Tehran Stock Exchange from 2012 to 2018. The obtained results showed that by controlling the interbank interest rate and the variety of bank loans and deposits, liquidity creation is significantly and directly associated with failure risk. Also, bank capital moderates this relationship, weakening the relationship between liquidity creation and failure risk. This result is consistent with the declarations of the Basel Committee, which always emphasize the role of the quantity and quality of bank capital in risk management. Accordingly, bankers and managers should pay more attention to the role and importance of this issue. Moreover, the results confirmed the insignificance of the monetary policy adopted by the central bank. This implies that the decisions made by the monetary authorities could be affected by some banking factors and become inefficient, regardless of the commitment to be implemented by the bank. For this reason, policymakers and monetary authorities must examine and study bank characteristics before making and implementing their decisions.

Finally, findings align with Berger & Bouwman (2009) on liquidity creation increasing risk, Acharya & Naqvi (2012) on capital moderating risk, and Diamond & Rajan (2001) on interbank contagion. Also, the results on monetary policy inefficiency contradict Ariccia et al. (2013) and Faiaa & Karau (2021), who highlight its role in risk management.

Recommendations

The present results point to the proficiency of bank capital in lowering the relationship between liquidity creation and failure risk, as well as the insignificance of the monetary policy adopted by the central bank. Consequently, other possible factors play a role in this relationship. Therefore, it can be a subject for future investigations to identify these factors and determine their effects. Thus, unlisted active banks should be investigated to provide supplementary research related to the current study. Also, other banking indices proposed in the CAMELS model for inter-bank comparisons can be considered in analyzing the role of capital. In addition, other exogenous macro variables, such as a crisis, should be used to study the role of monetary

policy.

In this paper, we analyze the ability of different machine learning algorithms to detect stock price jumps in Iran Khodro company. Two imbalanced data handling methods were introduced to tackle data imbalance problems: SMOTE and NearMiss. NearMiss showed better overall performance of the two methods as it had higher accuracy and a more balanced evaluation of performance metrics than SMOTE. Under the NearMiss method, CatBoost and Gradient Boosting proved to be the top-performing models in Precision and Recall. CatBoost was quite efficient, with an F1-score greater than 91%. However, the Random Forest model fared best using the SMOTE method, with a decent spillover between Precision and Recall. These findings are consistent with prior research, including studies from Prasad et al. (2021) and Gupta and Ahmed (2019) that second the merits of complex machine learning models, e.g., Gradient Boosting and CatBoost. The main difference in this study is that we use methods to handle imbalanced data and specialize in predicting final stock price jumps. In eliminating redundant samples using this NearMiss method, the balance between evaluation metrics was enhanced, enabling the effective separation of classes.

On the other hand, SMOTE raised recall by generating synthetic samples for the minority class but often adversely affected overall accuracy for multiple models. The objectives of this study were successfully met. The results show that combining machine learning with imbalanced data handling techniques substantially improves stock price jump prediction. Secondly, Iran Khodro stock data and focused analysis simultaneously filled existing gaps in the literature and served as a foundation for more accurate forecasting in the Iranian stock market.

Based on these results, this study can aid investors and financial analysts in making better decisions about risk management and allocation of financial resources by using optimal models like CatBoost and Gradient Boosting and imbalanced data handling techniques. These results may yield improved investment strategies and analytical tools that better forecast stock market fluctuations. There are, however, several limitations to this study. Finally, since the analysis builds on data inherently specific to Iran Khodro, its generalizability to other companies and financial markets might be questionable. For one, Iran Khodro was the subject of this study because it has the most significant trading volume and price fluctuations and is also greatly influenced by the Iranian stock market. As one of the biggest publicly traded companies, its stock price jump prediction can be used as a case study to

review stock price jump prediction.

Furthermore, these liquidity constraints and price fluctuation limits among all listed companies in the Iranian stock market imply that the insights gleaned from this study can extend to other stocks that operate under similar circumstances. Adding multiple companies may make the analyses more robust but remain highly relevant to market participants who examine stocks in a regulated financial environment. Second, the scope of the evaluation was intentionally limited due to computational constraints, thereby limiting the analysis of the models further. Lastly, external factors (economic and political changes affecting stock behavior) did not directly enter the modeling process. Furthermore, this study investigates the economic impact of prediction errors in stock price jump forecasting. In particular, this tradeoff between false negatives (missed jumps) and false positives (false alarms) is critical to financial decision-making. There is the risk of missing an actual price jump (false negative), especially in the Iranian stock market, where daily limits on price movements limit potential profit. However, a false optimistic prediction can lead to unnecessary transactions and capital misallocation. For example, investors focusing on maximizing profit might prefer models with higher recall, while risk-averse investors may prefer models maximized for precision. This insight contributes to the practical applicability of machine learning models in investment strategies, which enables traders to adjust to market conditions and risk tolerance.

The author of the study suggests future research:

New hybrid models that combine several data imputation methods usually used to handle multi-imbalanced datasets are explored to improve predictive performance.

- How can the combined effect of macroeconomic factors and company fundamentals be best investigated using machine learning techniques to increase predictive accuracy?
- Expand the scope of the analysis to cover other listed entities and industries to increase the extent of generalizability of the results.
- Further improve forecast accuracy by applying advanced deep learning such as Transformers.

This study showed that combining imbalanced data dealing techniques, including NearMiss and SMOTE, and sophisticated machine learning algorithms are effective methods for dealing with imbalanced data and improving price jump prediction. CatBoost was the best-performing model

based on stability and has a high balance across the evaluation metrics. This study also addressed research gaps and provided practical tools for more accurate stock market analysis for more rational decision-making. This research presents a novel framework for imbalanced data analysis and uses advanced machine learning algorithms to form existing research knowledge in stock market forecasting. In addition to applying SMOTE and NearMiss, this study presents a market-specific framework specific to the Iranian stock market's regulators' constraints and liquidity restrictions. Unlike existing work that applies imbalanced data management techniques in generalized settings, this research examines how stock price constraints (±5% daily volatility cap) affect data imbalance and predictive modeling in the Iranian financial market.

Furthermore, we incorporate three-dimensional PCA visualization to assess the effects of SMOTE and NearMiss on the data distribution, ensuring that synthetic and resampled data are financially interpretable and consistent with real market behavior. This addition can add practical value to resampling techniques in markets with structural limitations and fill a gap between algorithmic resampling and real-world financial constraints. Furthermore, the findings of this study can be a good reference point for future research in domestic and international financial markets.

Declaration of Conflicting Interests

The authors declared no potential conflicts of interest concerning the research, authorship and, or publication of this article.

Funding

The authors received no financial support for the research, authorship and, or publication of this article.

References

Arlot, S., & Celisse, A. (2010). A survey of cross-validation procedures for model selection. *Statistics Surveys*, *4*(none). https://doi.org/10.1214/09-ss054

Barreñada, L., Dhiman, P., Timmerman, D., Boulesteix, A., & Van Calster, B. (2024). Understanding overfitting in random forest for probability estimation: a visualization and simulation study. *Diagnostic and Prognostic Research*, *8*(1). https://doi.org/10.1186/s41512-024-00177-1

Bentéjac, C., Csörgő, A., & Martínez-Muñoz, G. (2020). A comparative analysis of gradient boosting algorithms. *Artificial Intelligence Review*, *54*(3), 1937–1967. https://doi.org/10.1007/s10462-020-09896-5

Bhamare, M., Kulkarni, P., Dholwani, D., Katyarmal, M., & Khatri, V. (2023). Prediction of Stock Market Closing Rates Using Deep Learning and Machine Learning Algorithms. *2023 IEEE 5th International Conference on Cybernetics, Cognition and Machine Learning Applications (ICCCMLA)*, *131-139.*, *11*, 131–139. https://doi.org/10.1109/icccmla58983.2023.10346622

Blanquero, R., Carrizosa, E., Ramírez-Cobo, P., & Sillero-Denamiel, M. R. (2021). Constrained Naïve Bayes with application to unbalanced data classification. *Central European Journal of Operations Research*, *30*(4), 1403–1425. https://doi.org/10.1007/s10100-021-00782-1

Chandar, S. K. (2024). Deep learning framework for stock price prediction using long short-term memory. *Soft Computing*. https://doi.org/10.1007/s00500-024-09836-3

Cosenza, D. N., Saarela, S., Strunk, J., Korhonen, L., Maltamo, M., & Packalen, P. (2024). Effects of model-overfit on model-assisted forest inventory in boreal forests with remote sensing data. *Forestry, an International Journal of Forest Research*. https://doi.org/10.1093/forestry/cpae055

Costa, V. G., & Pedreira, C. E. (2022). Recent advances in decision trees: an updated survey. *Artificial Intelligence Review*, *56*(5), 4765–4800. https://doi.org/10.1007/s10462-022-10275-5

Dorogush, A. V., Ershov, V., & Gulin, A. (2018). CatBoost: gradient boosting with categorical features support. *arXiv (Cornell University)*. https://doi.org/10.48550/arxiv.1810.11363

Elreedy, D., Atiya, A. F., & Kamalov, F. (2023). A theoretical distribution analysis of synthetic minority oversampling technique (SMOTE) for imbalanced learning. *Machine Learning*, *113*(7), 4903–4923. https://doi.org/10.1007/s10994-022-06296-4

Emami, S., & Martínez-Muñoz, G. (2023). Sequential training of neural networks with gradient boosting. *IEEE Access*, *11*, 42738–42750. https://doi.org/10.1109/access.2023.3271515

Gholami, N., & Shams Gharne, N. (2024). Presenting an Optimized CNN-LSTM

Model for Stock Price Forecasting in the Tehran Stock Exchange. Financial Management Perspective, 14(45), 123–147. doi: 10.48308/jfmp.2024.104892

Gu, S., Kelly, B., & Xiu, D. (2020). Empirical asset pricing via machine learning. *Review of Financial Studies*, *33*(5), 2223–2273. https://doi.org/10.1093/rfs/hhaa009

Gupta, V., & Ahmad, M. (2019). Stock price trend prediction with long short-term memory neural networks. International Journal of Computational Intelligence Studies, 8(4), 289. https://doi.org/10.1504/ijcistudies.2019.10025266

Hancock, J., & Khoshgoftaar, T. M. (2021). Leveraging LightGBM for Categorical Big Data. *2021 IEEE Seventh International Conference on Big Data Computing Service and Applications (BigDataService)*. https://doi.org/10.1109/bigdataservice52369.2021.00024

Heidari, M., & Amiri, H. (2022). Inspecting the Predictive Power of Artificial Intelligence Models in Predicting the Stock Price Trend in Tehran Stock Exchange. Financial Research Journal, 24(4), 602-623. doi: 10.22059/frj.2022.320064.1007149

Izsák, T., Marák, L., & Ormos, M. (2023). EVALUATION OF SUPPORT VECTOR MACHINE BASED STOCK PRICE PREDICTION. *Applied Computer Science*, *19*(3), 64–82. https://doi.org/10.35784/acs-2023-25

Jafarnejad Chaghoshi, A. , Rezasoltani, A. and Khani, A. M. (2024). Unleashing the Power of Ensemble Learning: Predicting National Ranks in Iran's University Entrance Examination. Industrial Management Journal, 16(3), 457-481. doi: 10.22059/imj.2024.381521.1008178

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2021). An introduction to statistical learning. In *Springer texts in statistics*. https://doi.org/10.1007/978-1-0716-1418-1

Johnson, N. F., Jefferies, P., & Hui, P. M. (2003). *Financial market complexity*. https://doi.org/10.1093/acprof:oso/9780198526650.001.0001

Khairi, T. W. A., Zaki, R. M., & Mahmood, W. A. (2019). Stock Price Prediction using Technical, Fundamental and News based Approach. *2019 2nd Scientific Conference of Computer Sciences (SCCS), 177–181.* https://doi.org/10.1109/sccs.2019.8852599

Khandagale, H. P., Patil, R., Patil, S., & Bhosale, D. (2023). PREDICTING STOCK PRICES WITH MACHINE LEARNING USING COMPARATIVE ANALYSIS OF RANDOM FOREST ALGORITHM. *International Journal of Engineering Applied Sciences and Technology*, *8*(6), 60–68. https://doi.org/10.33564/ijeast.2023.v08i06.008

Kleinbaum, D. G., & Klein, M. (2010). Logistic regression. In *Statistics in the health sciences*. https://doi.org/10.1007/978-1-4419-1742-3

Kotsiantis, S. B. (2011). Decision trees: a recent overview. *Artificial Intelligence*

Review, *39*(4), 261–283. https://doi.org/10.1007/s10462-011-9272-4

Li, Q., He, Y., & Pan, J. (2023). CrossFuse-XGBoost: accurate prediction of the maximum recommended daily dose through multi-feature fusion, cross-validation screening and extreme gradient boosting. *Briefings in Bioinformatics*, *25*(1). https://doi.org/10.1093/bib/bbad511

Liu, G., Chen, X., Luan, Y., & Li, D. (2024). VirusPredictor: XGBoost-based software to predict virus-related sequences in human data. *Bioinformatics*, *40*(4). https://doi.org/10.1093/bioinformatics/btae192

Lu, H., & Hu, X. (2023). Enhancing financial risk prediction for listed Companies: A CatBoost-Based Ensemble Learning Approach. *Journal of the Knowledge Economy*, *15*(2), 9824–9840. https://doi.org/10.1007/s13132-023-01601-5

Mathanprasad, L., & Gunasekaran, M. (2022). Analysing the Trend of Stock Marketand Evaluate the performance of Market Prediction using Machine Learning Approach. *2022 International Conference on Advances in Computing, Communication and Applied Informatics (ACCAI)*. https://doi.org/10.1109/accai53970.2022.9752616

Mehta, P., Pandya, S., & Kotecha, K. (2021). Harvesting social media sentiment analysis to enhance stock market prediction using deep learning. *PeerJ Computer Science*, *7*, e476. https://doi.org/10.7717/peerj-cs.476

Moerman, T., Santos, S. A., González-Blas, C. B., Simm, J., Moreau, Y., Aerts, J., & Aerts, S. (2018). GRNBoost2 and Arboreto: efficient and scalable inference of gene regulatory networks. *Bioinformatics*, *35*(12), 2159–2161. https://doi.org/10.1093/bioinformatics/bty916

Parashar, D., DSilva, M., & Kulshreshtha, S. (2023). A Machine Learning Framework for Stock Prediction using Sentiment Analysis. *2023 4th IEEE Global Conference for Advancement in Technology (GCAT), 1–5.*, *1*, 1–5. https://doi.org/10.1109/gcat59970.2023.10353541

Pflueger, C., Siriwardane, E., & Sunderam, A. (2020). Financial Market risk perceptions and the Macroeconomy*. *The Quarterly Journal of Economics*, *135*(3), 1443–1491. https://doi.org/10.1093/qje/qjaa009

Prasad, V. V., Gumparthi, S., Venkataramana, L. Y., Srinethe, S., Sree, R. M. S., & Nishanthi, K. (2021). Prediction of stock prices using statistical and machine learning models: A comparative analysis. *The Computer Journal*, *65*(5), 1338–1351. https://doi.org/10.1093/comjnl/bxab008

Rezaeyan, S., Taleghani, M., & Sherejsharifi, A. (2024). Developing a Comprehensive Model for Predicting Stock Prices in the Stock Market Using an Interpretive Structural Modeling Approach. Financial Research Journal, 26(3), 553–578. doi: 10.22059/frj.2023.364348.1007501

Sharif far, A., Khaliliaraghi, M., Raeesi Vanani, I., & Fallahshams, M. (2022). Application of Deep Learning Architectures in Stock Price Forecasting: A Convolutional Neural Network Approach. Journal of Asset Management and

Financing, 10(3), 1-20. doi: 10.22108/amf.2022.129205.1673

Shen, J., & Shafiq, M. O. (2020). Short-term stock market price trend prediction using a comprehensive deep learning system. *Journal of Big Data*, *7*(1). https://doi.org/10.1186/s40537-020-00333-6

Sornavalli, G., Angelin, G., & Khanna, N. H. (2022). Intelligent forecast of stock markets to handle COVID-19 economic crisis by modified generative adversarial networks. *The Computer Journal*, *65*(12), 3250–3264. https://doi.org/10.1093/comjnl/bxac056

Subekti, H., & Saepudin, D. (2022). Cross-Sectional Machine Learning Approach on Predicting Stock Return of LQ45 Index. *2022 1st International Conference on Software Engineering and Information Technology (ICoSEIT)*, *192-197.*, *2*, 192–197. https://doi.org/10.1109/icoseit55604.2022.10030044

Syriopoulos, P. K., Kalampalikis, N. G., Kotsiantis, S. B., & Vrahatis, M. N. (2023). kNN Classification: a review. *Annals of Mathematics and Artificial Intelligence*. https://doi.org/10.1007/s10472-023-09882-x

Tanha, J., Abdi, Y., Samadi, N., Razzaghi, N., & Asadpour, M. (2020). Boosting methods for multi-class imbalanced data classification: an experimental review. *Journal of Big Data*, *7*(1). https://doi.org/10.1186/s40537-020-00349-y

Wickramasinghe, I., & Kalutarage, H. (2020). Naive Bayes: applications, variations and vulnerabilities: a review of literature with code snippets for implementation. *Soft Computing*, *25*(3), 2277–2293. https://doi.org/10.1007/s00500-020-05297-6

Zhang, C., Zhou, X., & Wang, J. (2022). A financial risk early warning of listed companies based on PCA and BP Neural network. *Mobile Information Systems*, *2022*, 1–11. https://doi.org/10.1155/2022/8320329.

Bibliographic information of this paper for citing:

Jafarnejad, Ahmad; Rezasoltani, Arman & Khani, Amir Mohammad (2025). Comparative Analysis of Machine Learning Algorithms in Predicting Jumps in Stock Closing Price: Case Study of Iran Khodro Using NearMiss and SMOTE Approaches. *Iranian Journal of Finance*, 9(3), 27-54.

Copyright © 2025, Ahmad Jafarnejad, Arman Rezasoltani and Amir Mohammad Khani