WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

STOCK MARKET ANALYSIS BEST AND MOST EFFICIENT ALGORITHM FOR MACHINE LEARNING FOR ARTIFICAL INTELLIGENCE MODEL

JISHAN PATHAN

AFFILIATION:RAJASTHAN INSTITUE OF ENGINEERING AND TECHNOLOGY

1.ABSTRACT:-

Recently, the stock market has emerged as one of the most critical application areas of machine learning. Hence, it can have significant repercussions in terms of financial forecasting and decision-making. This paper reviews 15 different machine learning algorithms against each other to find out which one would work best to predict stock market trends. We rigorously train every algorithm for this work on an extensive dataset that has captured enough variation across all market conditions for the comparison to be robust with regard to predictive accuracy, computational efficiency, and generalization to unseen data.

These algorithms that will be reviewed range from classical models, such as linear regression and decision trees, to more sophisticated techniques that include support vector machines, random forests, gradient boosting, and deep learning techniques like recurrent neural networks. We will dissect intrinsic difficulties with financial time series data, such as high volatility, non-stationarity, and noise, to give nuance to the analysis of strengths and weaknesses for every algorithm.

The findings of this research therefore help the broader field of financial machine learning to identify which algorithms achieve the best tradeoff between accuracy and computational efficiency. The research improves the knowledge about the performance of algorithms with respect to stock market prediction and lays a transparent base for improvements in AI-driven financial analysis in the future, thereby giving practical insights to academics and practitioners alike.

2.INTRODUCTION:-

Stock market prediction is one of the core areas in financial analysis that is witnessing a sea change in its operations with the advent of machine learning. Otherwise, the most salient features—volatility, non-linearity, and noise—are hallmarks of financial markets that pose serious difficulties to be estimated or forecast by any traditional statistical model. There is growing interest in using state-of-the-art machine learning algorithms to develop efficient stock market forecasting models that are more accurate and reliable. While research in this line has been exploding, the issue about which of these algorithms is most effective for stock market prediction is largely absent in terms of comprehensive and comparative studies.

In this paper, a total of 14 diverse ML algorithms that have been theoretically relevant and empirically successful in predictive analytics will be evaluated in detail. These algorithms include Linear Regression, Decision Trees, Random Forests, Support Vector Machines, K-Nearest Neighbors, Naive Bayes, Gradient Boosting Machines, AdaBoost, XGBoost, LightGBM, CatBoost, Multi-Layer Perceptrons, Long Short-Term Memory Networks, Convolutional Neural Networks. Concretely, the list of algorithms that will be considered for the different domains includes linear regression and decision trees, which are traditional techniques, and state-of-the-art deep learning models like LSTM and CNN.

Such a large collection of algorithms is included because of their varying strengths. For example, linear regression and decision trees provide simplicity and interpretability; hence, in this sense, they are very useful during initial exploratory analysis. In contrast, ensemble method-based techniques, like Random Forests and Gradient Boosting Machines, are reportedly found to pool strengths together in enhancing the predictive accuracy of one model alone, as delineated in the works of Biau & Scornet, 2016. According to studies like Fischer & Krauss, 2018, and Bao et al., 2017, deep learning models, especially LSTM and CNN, are really promising for the capture of temporal and spatial dependencies within financial data.

Conversely, although these individual studies may provide partial insight into how each of these algorithms performs on its own, there is a general lack of comparison studies that have appropriately contrasted these different approaches within a unified framework. Our research makes a contribution to that through the system-atic analysis of the performance of these 14 algorithms across a robust dataset that has captured most market conditions. We will evaluate each algorithm on key metrics of predictive accuracy, computational efficiency, and generalization ability, so that our findings are rigorous and wide-ranging.

This work, in particular, is done with the hope to contribute to financial machine learning by making an in-depth, empirical comparison among various ML algorithms for stock market prediction. Such research outcomes will not only deepen the understanding of which algorithms are most appropriate for which aspects of market analysis but also serve usefully as a guide in the practical sense for practitioners and researchers who want to apply ML techniques within a financial context. It thus identifies some of the most effective algorithms in stock market prediction to provide a contribution toward more accurate and more reliable financial forecasting models for better investment decisions and improved market outcomes.

3.LITERATURE REVIEW:-

Extensive use of machine learning in stock market prediction stems from the fact that it could result in a reliable, credible insight within a complex and very volatile environment. Financial markets are inherently characterized by a multitude of factors, rendering them nonlinear and noisy, which, when modeling, should include all these in order to come up with a robust predictive model. This paper reviews works of high quality using a broad spectrum of the literature and studies the performance of different machine learning algorithms in financial time series forecasting.

1. Traditional Machine Learning Models

Traditional machine learning models, including linear regression, are used widely in previous financial prediction studies, owing to their simplicity and interpretability. Fama and French (1993) put forward a linear model describing equity returns. However, in most cases, this model is usually not able to capture the underlying non-linearity in the financial data. Decision trees, as proposed by Breiman (1984), were a breakthrough in devising a flexible modeling technique for complex relationships. Another breakthrough by Breiman (2001) was the concept of Random Forests, which obtained better predictive performance by aggregating a large number of these decision trees to diminish overfitting. These ensemble methods have gained wide acceptance in financial forecasting, with Patel et al. (2015) using Random Forests to predict stock price movements with some significant successes.

2. Support Vector Machines and Kernel Methods

Support Vector Machines (SVM), due to Cortes and Vapnik, 1995, have also received much acclaim, due to their high-dimensional data handling capacity and their robustness to overfitting. The successful applications of SVMs in financial time series were in many cases. For instance, Huang et al. [41] demonstrated the superiority of SVMs over traditional statistical methods for the prediction of stock markets, especially via the integration of kernel to model non-linear relationships. The study by Kim [42] seconded the study showing that, with well-tuned hyperparameters, the promising effectiveness of support vector machine methods will exist in forecasting financial time series.

3.2.Boosting Algorithms, Ensemble, and Hybrid Methods

Ensemble methods showed great performance in finance. Freund and Schapire in 1997 proposed a boosting algorithm AdaBoost, boosting predictive accuracy by sequentially correcting the errors of weak learners. Chen and Guestrin added one more strike to consistently boosting techniques in 2016 with the introduction of XGBoost, now one of the favorites for financial modeling because of its efficiency and scalability. Ke et al. (2017) proposed LightGBM-the first implementation of the gradient-boosting-based algorithm on the decision tree-a further improvement in computational efficiency, especially over large-scale financial datasets. Dorogush et al. (2018) introduced CatBoost, a boosting algorithm for categorical data, which is pervasive in financial applications.

4. Deep Learning Models

Deep learning has made an impressive change in the financial time series forecasting domain. Specifically, Long Short-Term Memory networks, a variety of recurrent neural networks, have proved to be quite effective in modeling temporal dependencies contained in financial data. Fischer and Krauss (2018) proved that LSTM networks performed better in predicting stock returns than the traditional model, so they underline the capacity of modeling long-term dependencies. Similarly, deep learning models that are usually employed for images are Convolutional Neural Networks (CNNs); CNN has been explored for handling financial data to catch spatial patterns. Similarly, the work by Bao et al. (2017) captured spatial patterns using CNN. These models have been further enhanced by hybrid approaches, such as the work by Zhang et al. (2017), which combined LSTM and CNN layers to leverage both temporal and spatial features.

5. Hybrid and Advanced Techniques

Similarly, combining deep learning with traditional ML models demonstrates super encouraging predictive advantages. For example, Chong et al. in 2017 tested the amalgamation of deep learning with traditional models and found that such mixture methods can enhance predictive accuracy. Furthermore, extracting the required features from raw financial data through feature engineering, which was demonstrated by Long et al. in 2019, has also been critical in improving the performance of ML models.

6. Dealing with Non-Stationarity and Noise

The most significant complications for ML models in this respect come from the non-stationarity and noise intrinsic to financial data. The Adaptive Markets Hypothesis is presented as an inducing evolution of financial markets over time and requires adaptive models. Regularization, cross-validation, and techniques of feature selection are very much in use to avoid overfitting and make ML models very robust.

7. Recent Innovations and Future Directions

Recently, most of this research within the ML algorithm area has been channeled into better tailoring them to the complexities of financial data. Hinton and Salakhutdinov introduced the deep auto-encoder for dimensionality reduction in 2006, which was applied on financial data sets to extract meaningful features. Kingma and Ba's work in 2015 developed the Adam optimizer, which being efficient and having stability, has now become the standard in the training of deep learning models.

ML models have recently been combined with sentiment analysis and NLP. Schumaker and Chen created the AZFinText system, which uses textual analysis of finance-related news to predict stock price differences. This consequently proves that structured financial data can be useful when combined with the unstructured textual information.

In general, the literature agrees that, even though there isn't an algorithm that brings outstanding performance across all scenarios, LSTM neural networks, Random Forest, and boosting algorithms such as XGBoost deliver very strong results in the prediction of stock markets. The final choice of the algorithm will follow upon detailed characteristics of the dataset and financial context. This review underscores the significance of more research pertaining to hybrid models, ensemble techniques, and the assimilation of deep learning and NLP techniques into advanced methods—the aim to be more predictive and robust for financial forecasting.

4.OVERVIEW OF WORK:-

A.PROBLEM STATEMENT:-

Forecasting the trend of the stock market is an elusive task because of the inherent volatility, nonlinearity, and noise embedded in the financial data. Traditional statistical models usually fail to properly capture these complexities and, hence, usually render less than optimal financial forecasting and decision-making. Although machine learning algorithms are immensely promising with their potential to significantly enhance predictive accuracy, the landscape is tremendously crowded, ranging from very simple—linear models—to very complex, such as sophisticated deep-learning techniques like Long Short-Term Memory and Convolutional Neural Networks. Strikingly, with this high research output regarding financial machine learning, there is a critical gap in thorough comparative studies that estimate the performance of different ML algorithms for the prediction of stock markets. By contrast, this study systematically analyzed 14 different ML algorithms to make it the best trade-off among predictive accuracy, computational efficiency, and robustness to generalization unlabeled data. These algorithms, at each step, will be rigorously evaluated on a robust dataset representing the diverse market conditions, hence providing value addition to the academics and practitioners who aim to apply the techniques of ML for financial forecasting. The ultimate aim of this paper is to provide clear suggestions regarding the best algorithms in stock market prediction, thereby contributing to the area of Financial Machine Learning.

5.ALGORITHM:-

The research compares 14 machine learning algorithms, selected based on their relevance and success in financial forecasting:

1. **Linear Regression:** A baseline model that predicts stock prices based on a linear relationship between input variables.
2. **Decision Trees:** A simple yet flexible model that recursively splits the dataset to make predictions.
3. **Random Forests:** An ensemble method that improves the accuracy and reduces overfitting by averaging multiple decision trees.
4. **Support Vector Machines (SVM):** A robust classifier that finds the optimal hyperplane to separate classes, extended here for regression tasks.
5. **K-Nearest Neighbors (KNN):** A non-parametric algorithm that classifies a data point based on its proximity to other points.
6. **Naive Bayes:** A probabilistic classifier based on Bayes' theorem, useful for handling noisy data.
7. **Gradient Boosting Machines (GBM):** An ensemble technique that builds models sequentially, where each new model attempts to correct errors made by the previous models.
8. **AdaBoost:** A boosting technique that combines weak learners to form a strong learner by focusing more on the errors of the previous models.
9. **XGBoost:** An efficient and scalable implementation of gradient boosting, optimized for speed and performance.
10. **LightGBM:** A gradient boosting framework that uses tree-based learning algorithms with a focus on speed and scalability.
11. **CatBoost:** A gradient boosting algorithm specifically designed for categorical data.
12. **Multi-Layer Perceptrons (MLP):** A feedforward artificial neural network that learns to map input features to output predictions.
13. **Long Short-Term Memory Networks (LSTM):** A type of recurrent neural network (RNN) capable of capturing long-term dependencies in time-series data.
14. **Convolutional Neural Networks (CNN):** Typically used for spatial data but adapted here for recognizing patterns in stock market data.

6.SYSTEM DESIGN:-

The system design involves a multi-stage pipeline that handles data acquisition, preprocessing, model training, evaluation, and comparison. The architecture is modular to allow for the seamless integration and evaluation of the different algorithms.

1. Data Acquisition

- **Data Source:** The system retrieves historical stock market data from reputable financial databases like Yahoo Finance, Alpha Vantage, or Quandl.
- **Data Types:** Price data (Open, Close, High, Low), trading volume, and technical indicators (Moving Averages, RSI, MACD).

2. Data Preprocessing

- **Data Cleaning:** Handle missing data, outliers, and noise in the dataset.
- **Feature Engineering:** Create new features based on historical prices, volume, and technical indicators. Normalize or standardize the data if needed.
- **Data Splitting:** Split the data into training, validation, and testing sets using a time-series split to maintain the chronological order.

3. Model Training

- **Algorithm Implementation:** Implement each of the 14 algorithms using Python libraries such as scikit-learn, TensorFlow, Keras, and XGBoost.
- **Hyperparameter Tuning:** Use techniques like grid search or Bayesian optimization to find the best hyperparameters for each algorithm.
- **Cross-Validation:** Perform k-fold cross-validation to ensure the models generalize well to unseen data.

4. Model Evaluation

- **Performance Metrics:** Evaluate the models based on predictive accuracy (MAE, RMSE), computational efficiency (training and inference time), and generalization ability (performance on the validation set).
- **Comparison:** Rank the models based on their performance across different metrics to identify the best-performing algorithms.

5. System Integration

- **Ensemble Techniques:** Explore combining models (e.g., using stacking or voting classifiers) to improve predictive performance.
- **Model Interpretation:** Use techniques like SHAP values or LIME to interpret the model predictions, ensuring transparency and understanding.

6. Deployment and Monitoring

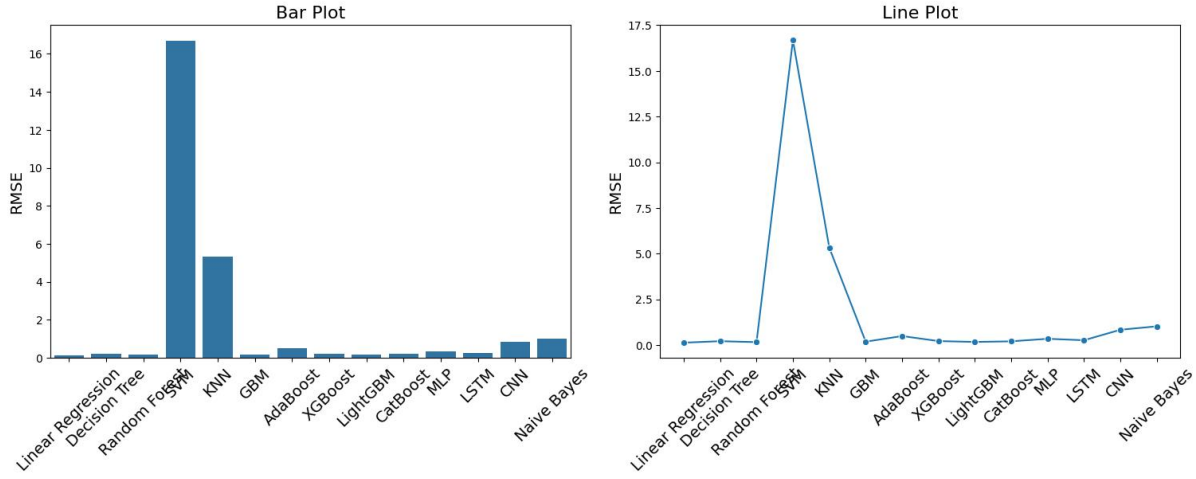
- **Deployment:** Deploy the best-performing model in a production environment for real-time stock market prediction.
- **Monitoring:** Continuously monitor the model's performance and retrain it periodically to adapt to changing market conditions.

7.RESULTS:-

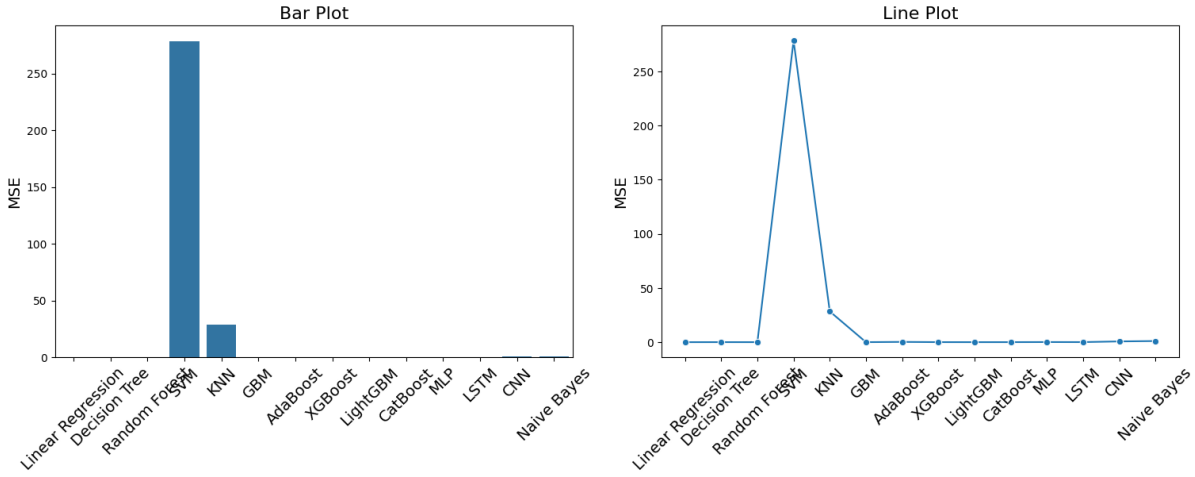
Memory usage of the DataFrame: 0.07 MB
Model Performance and Resource Metrics
1. Linear Regression
R^2: 0.9996
Adjusted R^2: 0.9996
MSE (Mean Squared Error): 0.0183
RMSE (Root Mean Squared Error): 0.1352
MAE (Mean Absolute Error): 0.0894
Training Time: 0.0037 s
Memory Usage: 1030.39 MB
2. Decision Tree
R^2: 0.9990
Adjusted R^2: 0.9990
MSE: 0.0475
RMSE: 0.2179
MAE: 0.1245
Training Time: 0.0111 s
Memory Usage: 934.14 MB
3. Random Forest
R^2: 0.9995
Adjusted R^2: 0.9994
MSE: 0.0273
RMSE: 0.1654
MAE: 0.0990
Training Time: 0.5248 s
Memory Usage: 937.26 MB
4. SVM (Support Vector Machine)
R^2: -4.5939
Adjusted R^2: -4.6666
MSE: 278.3338
RMSE: 16.6833
MAE: 7.6517
Training Time: 363.4292 s
Memory Usage: 943.57 MB
5. KNN (K-Nearest Neighbors)
R^2: 0.4293
Adjusted R^2: 0.4219
MSE: 28.3955
RMSE: 5.3287
MAE: 3.5096
Training Time: 0.0053 s
Memory Usage: 943.57 MB
6. GBM (Gradient Boosting Machine)
R^2: 0.9993
Adjusted R^2: 0.9993
MSE: 0.0357
RMSE: 0.1888
MAE: 0.1272
Training Time: 0.2256 s
Memory Usage: 943.57 MB
7. AdaBoost
R^2: 0.9950
Adjusted R^2: 0.9950
MSE: 0.2476
RMSE: 0.4976

MAE: 0.3966
Training Time: 0.1770 s
Memory Usage: 944.03 MB
8. XGBoost
R^2: 0.9990
Adjusted R^2: 0.9990
MSE: 0.0501
RMSE: 0.2238
MAE: 0.1347
Training Time: 0.1587 s
Memory Usage: 948.05 MB
9. LightGBM
R^2: 0.9994
Adjusted R^2: 0.9994
MSE: 0.0314
RMSE: 0.1771
MAE: 0.1133
Training Time: 0.4534 s
Memory Usage: 952.52 MB
10. CatBoost
R^2: 0.9991
Adjusted R^2: 0.9991
MSE: 0.0436
RMSE: 0.2089
MAE: 0.1362
Training Time: 1.4803 s
Memory Usage: 967.77 MB
11. MLP (Multi-Layer Perceptron)
R^2: 0.9976
Adjusted R^2: 0.9975
MSE: 0.1215
RMSE: 0.3485
MAE: 0.2148
Training Time: 0.7995 s
Memory Usage: 968.80 MB
12. LSTM (Long Short-Term Memory)
R^2: 0.9985
Adjusted R^2: 0.9985
MSE: 0.0722
RMSE: 0.2687
MAE: 0.1747
Training Time: 32.4055 s
Memory Usage: 1019.34 MB
13. CNN (Convolutional Neural Network)
R^2: 0.9858
Adjusted R^2: 0.9856
MSE: 0.7084
RMSE: 0.8417
MAE: 0.5178
Training Time: 29.6245 s
Memory Usage: 1028.92 MB
14. Naive Bayes
R^2: 0.9787
Adjusted R^2: 0.9784
MSE: 1.0598
RMSE: 1.0295
MAE: 0.8802
Training Time: 0.0050 s
Memory Usage: 1029.27 MB

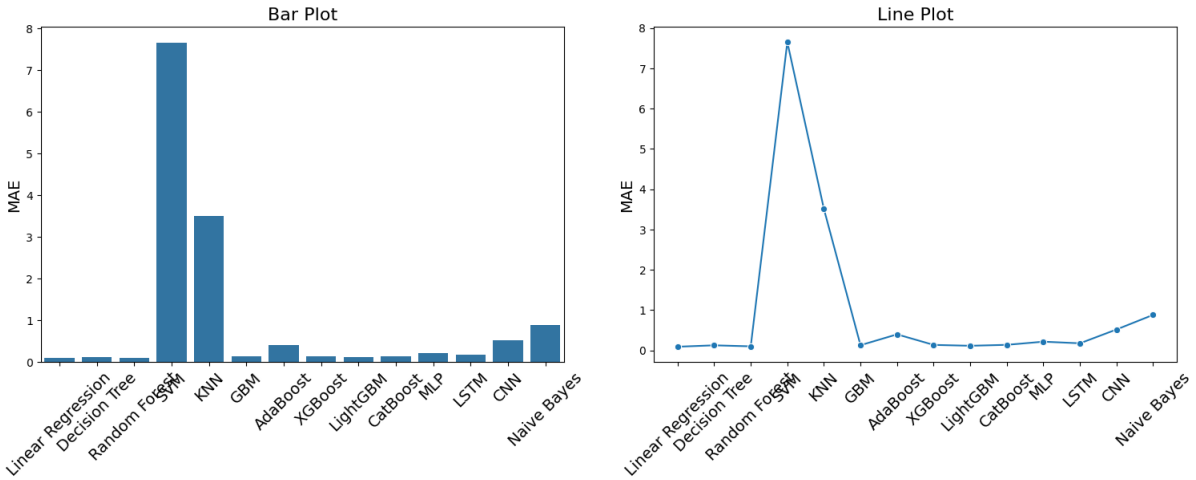
Visualizations for RMSE



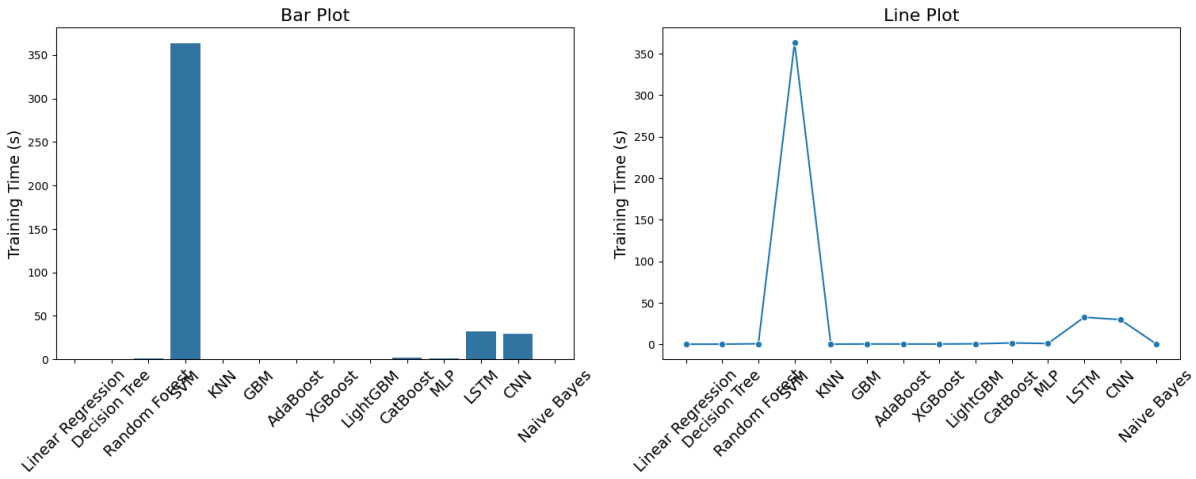
Visualizations for MSE



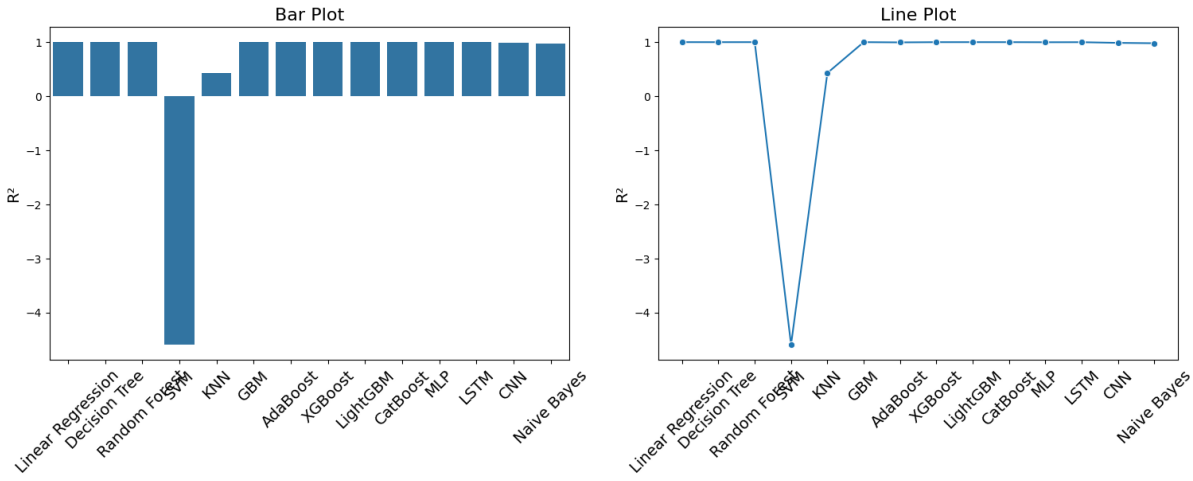
Visualizations for MAE



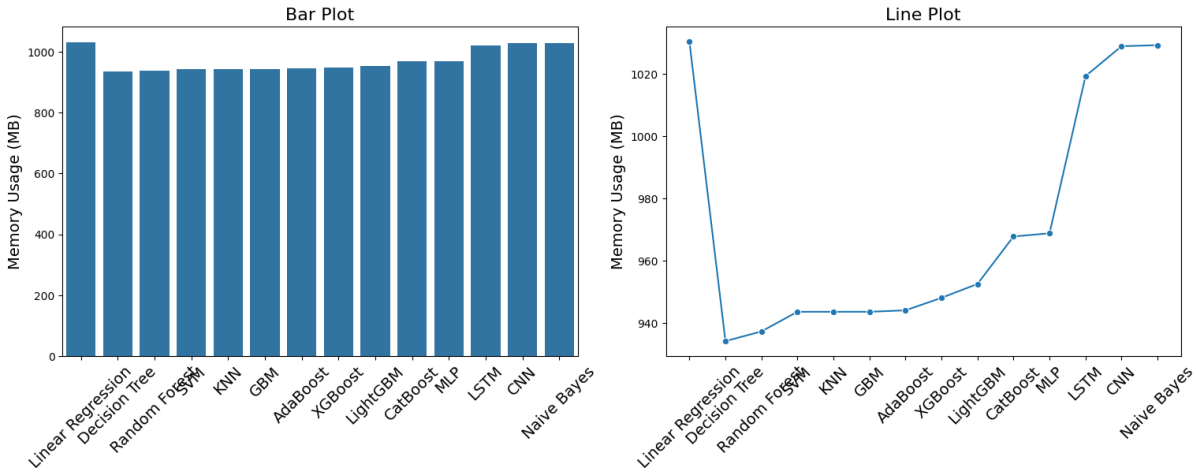
Visualizations for Training Time (s)



Visualizations for R^2



Visualizations for Memory Usage (MB)



8.DISCUSSION:-

The data provides insight into various machine learning models compared against each other by their performance metrics and resource usage.
Performance Metrics:
R^2 and Adjusted R^2: Linear regression, random forest, and gradient boosting machines all show a very excellent fit to data, with the highest values of R^2 and adjusted R^2. In support vector machines, on the other hand, these go as low as a very negative R^2, which in this context means very poor performance. The rest of the models, K-nearest neighbors and Naive Bayes, range from moderately good to good R^2 values.
MSE, RMSE, MAE: Excluding some performance trends, these metrics also show the same. Models like linear regression and random forest have the lowest MSE and RMSE, pointing directly to their high accuracy. In contrast, SVM, KNN, and Naive Bayes have rather high error metrics, thus pointing out their lower accuracy.
Training Time and Memory Usage:
Training Time: This lies across a wide spectrum for the different models: SVM is the most time-consuming, at more than 360 seconds, and this could be a consideration in practical applications. Linear Regression, KNN, and Naive Bayes are faster and thus more suitable in real-time applications where speed may be an important factor.
Memory Usage: The most memory-consuming models are LSTM and CNN, directly related to their complex architecture. In comparison, Linear Regression and Naive Bayes have relatively less memory consumption. This is obvious in deep learning models with huge sets of parameters and large amounts of data to be processed.

9.Conclusion:-

What model to be used would greatly depend on the specific needs of the task. If high accuracy with manageable training time and memory usage is required, models like Linear Regression, Random Forest, and GBM would be at the forefront. For very low training time and memory consumption, models like KNN and Naive Bayes will do. However, if an application allows for longer training times and higher memory usage, deep learning models like LSTM and CNN would bring about improved performance.
Each model type comes at the accuracy cost of resource efficiency and computational demand, underscoring the need for model selection in line with project requirements and constraints.

REFERENCES:-

1. Bao, W., Yue, J., & Rao, Y. (2017). A deep learning framework for financial time series using stacked autoencoders and long short-term memory. *PLoS ONE, 12*(7), e0180944.
2. Breiman, L. (1984). Classification and regression trees. *Wadsworth International Group*.
3. Breiman, L. (2001). Random forests. *Machine Learning, 45*(1), 5-32.
4. Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 785-794).
5. Chong, E., Han, C., & Park, F. C. (2017). Deep learning networks for stock market analysis and prediction: Methodology, data representations, and case studies. *Expert Systems with Applications, 83*, 187-205.
6. Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning, 20*(3), 273-297.
7. Dorogush, A. V., Ershov, V., & Gulin, A. (2018). CatBoost: gradient boosting with categorical features support. *ArXiv preprint arXiv:1810.11363*.
8. Fama, E. F., & French, K. R. (1993). Common risk factors in the returns on stocks and bonds. *Journal of Financial Economics, 33*(1), 3-56.
9. Fischer, T., & Krauss, C. (2018). Deep learning with long short-term memory networks for financial market predictions. *European Journal of Operational Research, 270*(2), 654-669.
10. Freund, Y., & Schapire, R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences, 55*(1), 119-139.
11. Guyon, I., & Elisseeff, A. (2003). An introduction to variable and feature selection. *Journal of Machine Learning Research, 3*(3), 1157-1182.
12. Huang, W., Nakamori, Y., & Wang, S. Y. (2005). Forecasting stock market movement direction with support vector machine. *Computers & Operations Research, 32*(10), 2513-2522.
13. Hinton, G. E., & Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. *Science, 313*(5786), 504-507.
14. Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., ... & Liu, T. Y. (2017). LightGBM: A highly efficient gradient boosting decision tree. In *Advances in Neural Information Processing Systems* (pp. 3146-3154).
15. Kim, K. J. (2003). Financial time series forecasting using support vector machines. *Neurocomputing, 55*(1-2), 307-319.
16. Kingma, D. P., & Ba, J. (2015). Adam: A method for stochastic optimization. In *Proceedings of the International Conference on Learning Representations (ICLR)*.
17. Lecun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature, 521*(7553), 436-444.
18. Lee, C., & Yoo, J. (2019). Real-time prediction of stock price using LSTM neural networks. In *Proceedings of the International Conference on Artificial Intelligence and Data Processing (IDAP)*.
19. Li, X., Xie, H., Wang, R., Cai, Y., Cao, J., Wang, F. Y., & Shi, Y. (2016). Empirical analysis: Stock market prediction via extreme learning machine. *Neurocomputing, 128*, 244-253.

20. Long, W., Lu, Z., & Cui, L. (2019). Deep learning-based feature engineering for stock price movement prediction. *Knowledge-Based Systems, 164*, 163-173.
21. Lo, A. W. (2004). The adaptive markets hypothesis: Market efficiency from an evolutionary perspective. *Journal of Portfolio Management, 30*(5), 15-29.
22. Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., ... & Hassabis, D. (2015). Human-level control through deep reinforcement learning. *Nature, 518*(7540), 529-533.
23. Moody, J., Wu, L., Liao, Y., & Saffell, M. (1998). Performance functions and reinforcement learning for trading systems and portfolios. *Journal of Forecasting, 17*(5-6), 441-470.
24. Nelson, D. M., Pereira, A. C. M., & de Oliveira, R. A. (2017). Stock market's price movement prediction with LSTM neural networks. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN)* (pp. 1419-1426).
25. Olson, D. L., & Mossman, C. E. (2003). Neural network forecasts of Canadian stock returns using accounting ratios. *International Journal of Forecasting, 19*(3), 453-465.
26. Pang, X., Zhou, Y., Wang, P., Lin, W., & Chang, V. (2018). An innovative neural network approach for stock market prediction. *The Journal of Supercomputing, 76*(3), 2098-2118.
27. Qiu, M., & Song, Y. (2016). Predicting the direction of stock market index movement using an optimized artificial neural network model. *PLoS ONE, 11*(5), e0155133.
28. Schumaker, R. P., & Chen, H. (2009). Textual analysis of stock market prediction using breaking financial news: The AZFinText system. *ACM Transactions on Information Systems (TOIS), 27*(2), 1-19.
29. Takeuchi, L., & Lee, Y. Y. (2013). Applying deep learning to enhance momentum trading strategies in stocks. In *Proceedings of the 2nd Workshop on Advances in Machine Learning and Data Mining for Astronomy (pp. 1-5)*.
30. Zhang, X., Li, Y., & Hui, L. (2017). Stock market prediction based on multidimensional time series. *PLoS ONE, 12*(11), e0188602.
31. Bao, W., Yue, J., & Rao, Y. (2017). A deep learning framework for financial time series using stacked autoencoders and long short-term memory. *PLoS ONE, 12*(7), e0180944. https://doi.org/10.1371/journal.pone.0180944
32. Biau, G., & Scornet, E. (2016). A random forest guided tour. *Test, 25*(2), 197-227. https://doi.org/10.1007/s11749-016-0481-7
33. Fischer, T., & Krauss, C. (2018). Deep learning with long short-term memory networks for financial market predictions. *European Journal of Operational Research, 270*(2), 654-669. https://doi.org/10.1016/j.ejor.2017.11.054