**ECE 579 Intelligent Systems, Winter 2024**

**Final Project Report**

**MNCs Stock Market Close Price Detection**

**Names of the students in the group:** Devyani Deore, Sai Sanjith Sivapuram

**Responsibilities of each student:**

1. **Devyani Deore:** Devyani would be involved in Data Engineering, Data Pre-Processing, and partly in the Exploratory Data Analysis and Modeling and Evaluation. It would be a combination of understanding the data by performing analysis of the structure and the features transformation, and cleaning data wherever necessary**.** Devyani would also perform one of the models and obtain the scores for the model implemented.
2. **Sai Sanjith Sivapuram**: Sanjith is partly involved in the Data Pre-Processing while performing the other necessary steps like Exploratory Data Analysis to understand the features and behavior of the data. Sanjith also performs appropriate normalization, feature extraction, and modeling on the data and retrieves evaluation metrics for the test data.

The members will be equally involved in the documentation and presentation of the project.

**Introduction:**

* Our focus will be on applying various predictive models to stock market data, primarily from multinational corporations (MNCs) named IBM, and then comparing the performances of the different models. Our approach involves the examination and analysis of diverse stock market datasets from MNCs.
* We aim to understand how different models work and explore additional techniques that could be applied to the data. This includes normalization methods and data scaling techniques, all to enhance the accuracy of predictions on the dataset.
* Pandas, Matplotlib, sci-kit-learn, Random Forest Regressor, Long Short-Term Memory (LSTM), and Autoregressive Integrated Moving Average (ARIMA) are certain technologies and libraries involved.
* We use the Min Max Scaler technique instead of the K-fold technique for normalization and to get better accuracies of the model from the data.
* We check the RMSE metric for all the models to conclude the best based on the scores.

**Description of Technologies Related to the Project:**

1. **Random Forests:**

* Random forest models build on multiple decision trees and enable us to retrieve more accurate forecasts from the data.
* This helps us to get more combinations enabling more information gain from the data.
* These models help us to understand the non-linear relations in the data.

1. **ARIMA (Autoregressive Integrated Moving Average):**

* ARIMA is well known for working with time series data (preferably stationary data).
* As our dataset contains the history of the stock prices for different companies hence the data is stationary and suitable for models like ARIMA to perform well.

1. **LSTM (Long Short-Term Memory) Neural Network:**

* LSTM models are based on RNN (recurrent neural networks) and work well with time series data.
* The main advantage of LSTM is its ability to hold long-term dependencies on the data which is crucial for our data as we have data of around 5 years and models like LSTM take the whole data under consideration.
* These models also handle non-linear relationships well.

1. **Decision Trees:**

* Decision Trees are known to work well with both regression and classification types of data.
* It helps to get information gain and relations between the features in the dataset enabling more accurate predictions.

We have decided to move forward with Random Forest Regressor, LSTM, and ARIMA for our dataset and compare the results from the models.

**Methods used in the project:**

Below are the steps we have performed to execute the mentioned models:

1. **Random Forest Regressor:**

* We have initiated the model using the sci-kit-learn library and provided the parameters such as the 200 estimators which define the number of decision trees the model should use for the ensemble approach and a random state value of 40 for different execution possibilities.
* As close price was our target variable, we compared the actual value with the predicted values for the future dates that are present in the test dataset.
* We have retrieved the RMSE value to compare the scores and obtained a score of 0.63.

A graph showing the price of a stock market

Description automatically generated

**Fig 1: Line plot for Actual vs Predicted Close Prices for IBM dataset using Random Forest**

1. **ARIMA:**

* Before getting started with the ARIMA model we installed the required libraries and performed certain pre-requisite checks on the dataset such as stationarity, rolling mean, and rolling standard deviation on the target variable.
* We have observed that the rolling mean and standard deviation are not stationary and have changed the data to become stationary by removing trends in the dataset and creating a train test split for the data.
* Check for the best parameters to put for the ARIMA model using the concept of auto Arima and evaluate the model which gave an RMSE score of 0.083.

A graph showing the price of a stock price

Description automatically generated

**Fig 2: Line plot for Actual vs Close Prices for IBM dataset using ARIMA**

1. **LSTM:**

* We have used the same train test split data for performing modeling and evaluations.
* We have defined an LSTM architecture using the Keras library. The model we have defined contains 4 LSTM layers with the ability to pass the outputs with the timestamps to the next layer in the architecture.
* For evaluation criteria, we have used RMSprop optimizer and mean squared error as the loss function to retrieve the scores obtained by the model.
* The model is trained on 50 epochs and a batch size of 32 and we obtained a RMSE score of 3.960.

A graph showing the price of a stock price

Description automatically generated

**Fig 3: Line plot for Actual vs Predicted Close Prices for IBM dataset using LSTM**

**Experiments performed on the project:**

* The data that we have chosen to use in this project is retrieved from Kaggle. It contains features such as Date, Open Price, Close Price, High, and Low for a given date.
* We experimented with finding the issues in the data like null values, and outliers, and pre-processed them accordingly.

**A screenshot of a computer

Description automatically generatedA screen shot of a computer program

Description automatically generated**

**Fig 4: Pseudocodes for Checking null values**

**A screenshot of a computer

Description automatically generated**

**Fig 5: Check the statistics for the features in the dataset**

* **Exploratory Data Analysis:** Using the concepts of data visualization enabled us to understand the patterns in the data even better.

A graph showing the growth of the ibm stock market

Description automatically generatedA graph showing the growth of the ibm stock market

Description automatically generated

**Fig 6: Line plot patterns of the Open and Close prices of the IBM stock dataset**

* As the data is more focused on time series analysis, we have sorted the data frame based on the Date feature to be able to split the data into train and test for modeling and evaluation.
* We have used Min Max Scaler techniques to normalize the data magnitude which led to the models performing better.

**Conclusion**

We have performed a total of 3 different regression and time-series-specific models on the IBM stock dataset. To evaluate the best model, we have chosen to use the RMSE error metric, and the one with the lowest score was found to be the ARIMA model with the lowest score of 0.083. To achieve the best accuracies possible, we have performed the required pre-processing and normalization techniques like scaling for the best data. Additionally, we also have performed the required data analysis to understand the patterns and outliers in the dataset.

During the whole project execution, we as a team learned about the ARIMA and LSTM models and how to implement them. Also, we had a good understanding of how to deal with time series data mainly we got to know about the importance of dates and their essence in train test splitting.

**References**

1. Moghar, A., & Hamiche, M. (2020). Stock market prediction using LSTM recurrent neural network. *Procedia computer science*, *170*, 1168-1173.
2. Polamuri, S. R., Srinivas, K., & Mohan, A. K. (2019). Stock market prices prediction using random forest and extra tree regression. *Int. J. Recent Technol. Eng*, *8*(1), 1224-1228.
3. Yadav, A., Jha, C. K., & Sharan, A. (2020). Optimizing LSTM for time series prediction in Indian stock market. *Procedia Computer Science*, *167*, 2091-2100.
4. Ariyo, A. A., Adewumi, A. O., & Ayo, C. K. (2014, March). Stock price prediction using the ARIMA model. In *2014 UKSim-AMSS 16th international conference on computer modelling and simulation* (pp. 106-112). IEEE.
5. Nti, K. O., Adekoya, A., & Weyori, B. (2019). Random forest based feature selection of macroeconomic variables for stock market prediction. *American Journal of Applied Sciences*, *16*(7), 200-212.