Submission for the Final Examination of EE4708- Data Analytics Lab

Jeshlin Donna J
Department of Metallurgical and Materials Engineering
Indian Institute of Technology Madras
Chennai, India
mm20b029@smail.iitm.ac.in

***Abstract*—This paper consists of two major divisions- the first part is a new version and perspective and exploratory analysis of the six assignments as part of this course. And the second division involves the analysis, insight derivation and model building for a dataset provided from the field of finance.**

***Index Terms*—Finance, Exploratory Analysis, Model Building**

I. ASSIGNMENT 1- ANALYZING THE RELATIONSHIP BETWEEN CANCER DATA AND SOCIOECONOMIC FACTORS

The second version of this assignment involved intensive pre-processing and using trial-and error method to find data models that best fit the data and help in the prediction of annual incidence and mortality rates.

The correlation between various features was identified and later visualized using a heatmap. The features having correlation greater than 0.9 with other features were removed to reduce redundancy in our model. This enables the machine learning algorithm to train faster. It reduces the complexity of any model on the data and makes it easier to interpret. It improves the accuracy of the model if the right subset is chosen.

It was found that the values of few features were significantly higher in magnitude than other features. In the machine learning algorithms if the values of the features are closer to each other there are chances for the algorithm to get trained well and faster instead of the data set where the data points or features values have high differences with each other will take more time to understand the data and the accuracy will be lower. The machine learning models also provide weights to the input variables according to their data points and inferences for output. In that case, if the difference between the data points is so high, the model will need to provide the larger weight to the points and in final results, the model with a large weight value is often unstable. This means the model can produce poor results or can perform poorly during learning. Due to this reason, feature scaling was performed on the data with the help of Standard Scalar from the ”sklearn” library to prevent unnecessary instability in our models.

Different machine learning models were tried out on the dataset to predict annual cancer incidence and mortality rates. Linear Regression, K-Neighbors Regression, Decision Tree Regression and Random Forest Regression models were trained on the dataset and used to predict the desired output features. The accuracy of these machine learning algorithms (Fig.1.) were inferred on the validation data set apart(not used during training). A basic deep learning algorithm was also deployed on the data (sequential model with multiple dense layers) and a K-fold evaluation was used in the same.

MODEL	RMSE LOSS ON VALIDATION DATA
LINEAR REGRESSION (VERSION 1)	0.66
LINEAR REGRESSION (VERSION 2)	0.46
K-NEIGHBORS REGRESSION (VERSION 2)	0.55
DECISION TREE REGRESSION (VERSION 2)	0.51
RANDOM FOREST REGRESSION (VERSION 2)	0.49
BASIC ANN MODEL (VERSION 2)	0.46

Fig. 1. RMSE Error Values for version-1 and different models under version-2 of assignment 1

The version 1 of this assignment had an RMSE(Root Mean Squared Error) of about 0.66 whereas the version 2 of this assignment yielded considerably better results with a RMSE of about 0.46.

II. ASSIGNMENT 2- ANALYZING SURVIVAL IN THE TITANIC SHIPWRECK

The second version of this assignment involved data manipulation techniques and trying out different data models on the given dataset and finding the model that best fits the data.

The correlation between various features were identified and visualized using a heatmap. The features having correlation greater than 0.9 with other features were removed to reduce redundancy in our model. Feature scaling was performed on the data with the help of Standard Scalar from the ”sklearn” library to prevent unnecessary instability in our machine

learning models.

Different machine learning models were tried out on the dataset to predict whether a particular passenger onboard would survive the shipwreck or not based on the information set of the passenger. 'Logistic Regression','Support Vector Machines','Linear Support Vector Machines', 'Decision Tree','Random Forest','k-Nearest Neighbours', 'Stochastic Gradient Descent', 'Perceptron' and 'Naive Bayes' models were trained on the dataset and used to predict the desired output feature- whether a passenger would survive or not. The accuracy of these machine learning algorithms were inferred and tabulated(Fig.2.). The highest accuracies were got by the Decision Tree and Random Forest models. But the final best fitting model was chosen to be the random forest model as its chances of overfitting are much lesser compared to the decision tree models.

Models	
Decision Tree	99.72
Random Forest	99.72
Naive Bayes	82.02
Logistic Regression	81.74
k-Nearest Neighbours	76.40
Stochastic Gradient Descent	71.07
Support Vector Machines	68.96
Perceptron	68.40
Linear Support Vector Machines	66.99

Fig. 2. Accuracy Scores for different models under version-2 of assignment-2

The version 1 of this assignment had an accuracy of about 81% whereas the version 2 of this assignment yielded a significantly higher accuracy of about 99%.

III. ASSIGNMENT 3- DETERMINING IF A PERSON MAKES OVER $50K ANUALLY

The second version of this assignment involved feature engineering and trail and error of different data models on the given dataset and finding the model that best fits the data extracted from the 1994 Census bureau database by Ronny Kohavi and Barry Becker (Data Mining and Visualization, Silicon Graphics).

The raw dataset was found to be having vague column/variable names, these variables were renamed with meaningful variable names after performing a background research on the dataset. The categorical variable ' <=50K' was mapped to numerical binary values (i.e. 0/1) to create a new binary variable '>50K'. Few rows were found to be containing meaningless/missing data, these rows were removed from the dataframe. New variables called "Working hour level", "capital_gain", "capital_loss" and "Native Region" have been created to better capture the data.

In the preprocessing stage, One-Hot Encoding was performed on the categorical variables 'Work Class', 'Education', 'Marital status', 'Occupation', 'Relationship', 'Race', 'Sex', 'Native Region', 'capital_gain' and 'capital_loss'. It was observed that the numeric columns in our dataset have varying ranges. Thus, the numerical variables- 'Age', 'fnlwgt', 'Years of education', '>50K' and 'Working hour level' were scaled using the Min-Max Scaler from the Ski-kit Learn library. Such scaling of the numeric features ensures that no particular feature has a disproportionate impact on the model's loss. The correlation between the variables "Working hours per week" and "Working hour level" was found to be very high. Hence the former variable was dropped before training to remove redundancy.

Different machine learning models were tried out on the dataset to predict whether a particular person earns over $50k per year or not based on different features of information of the person. 'Logistic Regression','Support Vector Machines','Linear Support Vector Machines', 'Decision Tree','Random Forest','k-Nearest Neighbours', 'Stochastic Gradient Descent', 'Perceptron' and 'Naive Bayes' models were trained on the dataset and used to predict the desired categorical output column. The accuracy of these machine learning algorithms on the validation dataset were inferred and tabulated(Fig.3.). The highest accuracy was achieved by the Stochastic Gradient Descent(SGD) model.

Models	
Stochastic Gradient Descent	85.29
Logistic Regression	85.22
Support Vector Machines	85.16
Linear Support Vector Machines	85.09
Random Forest	84.16
k-Nearest Neighbours	82.77
Decision Tree	79.52
Perceptron	77.67
Naive Bayes	67.00

Fig. 3. Accuracy Scores for different models on the validation data under version-2 of assignment-3

The version 1 of this assignment had an accuracy of about 67.04% whereas the version 2 of this assignment yielded a significantly higher accuracy of about 85.29%.

IV. ASSIGNMENT 4- EVALUATION OF CUSTOMER ACCEPTABILITY OF CARS(BASED ON DECISION TREE)

The second version of this assignment involved data preprocessing and trying out different data models on the given data and finding the model that best fits it.

Categorical variables were created from the given variables of 'object' type. Since all the categories in the variables had a naturally increasing order, they were converted to numbers (e.g. 1, 2, 3, 4, 5) preserving the order. These are called ordinals. The correlation between no variables were found to be significant, hence there was no requirement to drop any variable before training to remove redundancy.

The dataset was split into the training and validation parts and the training dataset was used to train different models to fit the given data. Different machine learning models like 'Logistic Regression','Support Vector Machines','Linear Support Vector Machines', 'Decision Tree','Random Forest','k-Nearest Neighbours', 'Stochastic Gradient Descent', 'Perceptron' and 'Naive Bayes' models were trained on the dataset and used to predict the customer acceptability of cars given their data. The accuracy of these machine learning algorithms on the validation dataset were inferred and tabulated(Fig.4.). The highest accuracy was found to be achieved by the Decision Tree model.

Models	
Decision Tree	97.45
Random Forest	96.99
Support Vector Machines	96.53
k-Nearest Neighbours	90.74
Logistic Regression	81.48
Linear Support Vector Machines	78.24
Stochastic Gradient Descent	77.31
Perceptron	75.46
Naive Bayes	68.52

Fig. 4. Accuracy Scores for different models on the validation data under version-2 of assignment-4

The decision tree model achieved an accuracy of 100% on the training dataset but a accuracy of about 97% on the validation data. Hence, operations were performed to check if the model was over-fitting the data. Plots (Fig.5.) were made between the Training and Validation accuracy with hyper-parameters of a decision tree model like maximum depth, maximum number of nodes etc. And it was eventually concluded that the model was not over-fitting and that the model had already achieved its maximum potential accuracy and already generalized to its best.

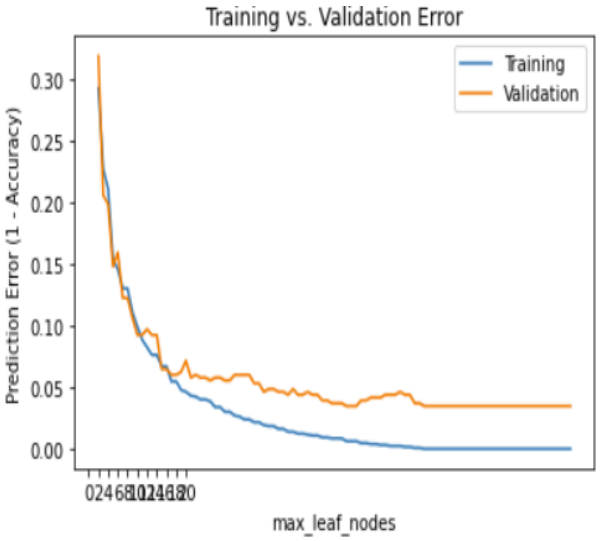


Fig. 5. Training vs Validation Error for different values of maximum number of leaf nodes for the decision tree model

V. ASSIGNMENT 5- EVALUATION OF CUSTOMER ACCEPTABILITY OF CARS(BASED ON RANDOM FOREST)

The second version of this assignment involved data preprocessing and trying out different data models on the given data and finding the model that best fits it.

Categorical variables were created from the given variables of 'object' type. Since all the categories in the variables had a naturally increasing order, they were converted to numbers preserving the order, called ordinals. The correlation between no variables were found to be significant, hence there was no requirement to drop any variable before training to remove redundancy.

The dataset was split into the training and validation parts and the training dataset was used to train different models to fit the given data. Different machine learning models like 'Logistic Regression','Support Vector Machines','Linear Support Vector Machines', 'Decision Tree','Random Forest','k-Nearest Neighbours', 'Stochastic Gradient Descent', 'Perceptron' and 'Naive Bayes' models were trained on the dataset and used to predict the customer acceptability of cars given their data. The accuracy of these machine learning algorithms on the validation dataset were inferred and tabulated(Fig.4.).

The random forest model achieved an accuracy of 100% on the training dataset but a accuracy of about 96.99% on the validation data. Hence, operations were performed to check if the model was over-fitting the data. Plots (Fig.5.) were made

between the Training and Validation accuracy with hyper-parameters of the random forest model(number of estimators etc..). And it was eventually concluded that the random forest model was not over-fitting and that the model had already achieved its maximum potential and had generalized to its maximum potential.

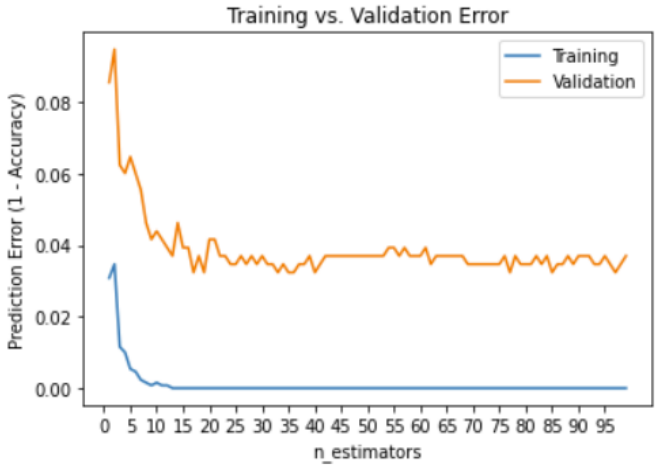


Fig. 6. Training vs Validation Error for different values of number of estimators for the random forest model

VI. ASSIGNMENT 6- IDENTIFICATION OF PULSAR STARS

The second version of this assignment involved verious data preprocessing techniques and trying out different data models on the given dataset and finding the model that best fits the given dataset.

The variables- 'Mean of the integrated profile', 'Standard deviation of the integrated profile', 'Excess kurtosis of the integrated profile', 'Skewness of the integrated profile', ' Mean of the DM-SNR curve', 'Standard deviation of the DM-SNR curve', 'Excess kurtosis of the DM-SNR curve', 'Skewness of the DM-SNR curve' were taken as the continuous input variables and were used to predict the binary categorical output variable-'target_class'. It was found that the values of few features were significantly higher in magnitude than other features. Machine learning models also provide weights to the input variables according to their data points and inferences for output. In that case, if the difference between the data points is so high, the model will need to provide the larger weight to the points and in final results, the model with a large weight value is often unstable. This means the model can produce poor results or can perform poorly during learning. Due to this reason, feature scaling was performed on the data with the help of Standard Scalar from the "sklearn" library to prevent any kind of instability in our models.

Different machine learning models were tried out on the dataset to predict whether a particular candidate was a pulsar star or not based on the information set of the candidate. 'Logistic Regression','Support Vector Machines','Linear Support Vector Machines', 'Decision Tree','Random Forest','k-Nearest Neighbours', 'Stochastic Gradient Descent', 'Perceptron' and 'Naive Bayes' models were trained on the dataset and used to predict the desired output feature. The accuracy of these machine learning algorithms were inferred and tabulated(Fig.7.). The highest accuracy on the validation dataset was achieved on the Logistic Regression model.

Models	
Logistic Regression	97.80
Random Forest	97.80
Stochastic Gradient Descent	97.63
Support Vector Machines	97.59
k-Nearest Neighbours	97.59
Perceptron	97.54
Decision Tree	96.68
Naive Bayes	95.21
Linear Support Vector Machines	89.65

Fig. 7. Accuracy Scores for different models under version-2 of assignment-6

The version 1 of this assignment had an accuracy of about 97.59% whereas the version 2 of this assignment yielded a mildly but recognizably higher accuracy of about 97.8%.

VII. FINAL EXAMINATION QUESTION

A. INTRODUCTION

The data considered was the Open, Close, High, Low and Volume Data of Stocks/Exchanges of 7 different kinds. The key task was the performing of Exploratory data analysis to uncover previously unknown or hidden facts in the data set available, visualize them using Data Visualization and finally make predictions on the validation data using the most apt data model(which was found out to be "Long Short Term Mermory" (LSTM).

Long Short Term Memory networks – usually just called "LSTMs" – are a special kind of RNN, capable of learning long-term dependencies. They work tremendously well on a large variety of problems, and are widely used.Long short-term memory (LSTM) is an artificial recurrent neural network (RNN) architecture used in the field of deep learning. Unlike standard feedforward neural networks, LSTM has feedback connections. It can process not only single data points (such as images), but also entire sequences of data (such as speech or video or time series data). A common

LSTM unit is composed of a cell, an input gate, an output gate and a forget gate. The cell remembers values over arbitrary time intervals and the three gates regulate the flow of information into and out of the cell. LSTM networks are well-suited to classifying, processing and making predictions based on time series data, since there can be lags of unknown duration between important events in a time series. LSTMs were developed to deal with the vanishing gradient problem that can be encountered when training traditional RNNs.

The primary goal of this section is to perform extensive exploratory analysis on the data and to reveal interesting and useful insights. The secondary goal is to build a model best fitting to the data that will be used to predict the stock close prices on several days. A predictive model based on LSTM was built using the given data so that the stock high, low, open, close and volume features of the time series data contributed to the prediction of the stock close prices on a particular day.

In this paper, we conducted exploratory data analysis to excavate different knowledge existing in the given data set and to perceive the impact of every field on one another. Data pre-processing was done, data analysis was performed and, a predictive LSTM model was built and trained on the training data, and the accuracy was tested on the validation dataset. After analyzing the dataset, we discovered insights and information on the various trends and outliers found in the stock data of various stock/exchange kinds

B. Long short-term memory (LSTM)

Recurrent Neural Networks suffer from short-term memory. If a sequence is long enough, they have a hard time carrying information from earlier time steps to later ones. During back propagation, recurrent neural networks suffer from the vanishing gradient problem. Gradients are values used to update a neural networks weights. The vanishing gradient problem is when the gradient shrinks as it back propagates through time. If a gradient value becomes extremely small, it doesn't contribute too much learning. In theory, classic (or "vanilla") RNNs can keep track of arbitrary long-term dependencies in the input sequences. The problem with vanilla RNNs is computational (or practical) in nature: when training a vanilla RNN using back-propagation, the long-term gradients which are back-propagated can "vanish" (that is, they can tend to zero) or "explode" (that is, they can tend to infinity), because of the computations involved in the process, which use finite-precision numbers. RNNs using LSTM units partially solve the vanishing gradient problem, because LSTM units allow gradients to also flow unchanged.

LSTM 's were created as the solution to short-term memory. They have internal mechanisms called gates that can regulate the flow of information. These gates can learn which data in a sequence is important to keep or throw away. By doing that, it can pass relevant information down the long chain of sequences to make predictions.

An LSTM has a similar control flow as a recurrent neural network. It processes data passing on information as it propagates forward. The differences are the operations within the LSTM's cells.

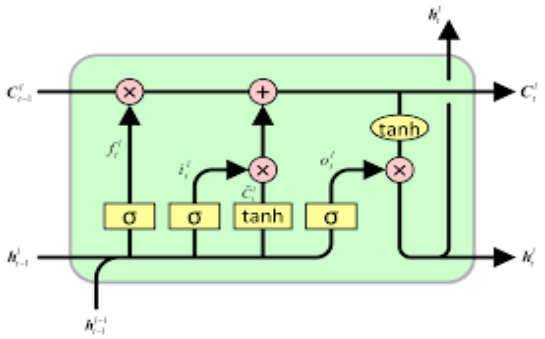


Fig. 8. LSTM Cell and It's Operations

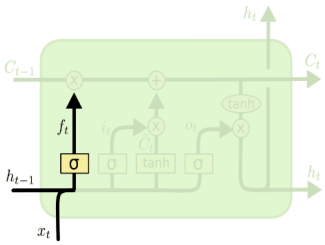
These operations are used to allow the LSTM to keep or forget information.

1) Core Concept: The core concept of LSTM's are the cell state, and it's various gates. The cell state act as a transport highway that transfers relative information all the way down the sequence chain. You can think of it as the "memory" of the network. The cell state, in theory, can carry relevant information throughout the processing of the sequence. So even information from the earlier time steps can make it's way to later time steps, reducing the effects of short-term memory. As the cell state goes on its journey, information get's added or removed to the cell state via gates. The gates are different neural networks that decide which information is allowed on the cell state. The gates can learn what information is relevant to keep or forget during training.

2) Sigmoid: Gates contains sigmoid activations. A sigmoid activation is similar to the tanh activation. Instead of squishing values between -1 and 1, it squishes values between 0 and 1. That is helpful to update or forget data because any number getting multiplied by 0 is 0, causing values to disappears or be "forgotten." Any number multiplied by 1 is the same value therefore that value stay's the same or is "kept." The network can learn which data is not important therefore can be forgotten or which data is important to keep.

3) Input Gate: To update the cell state, we have the input gate. First, we pass the previous hidden state and current input into a sigmoid function. That decides which values will be updated by transforming the values to be between 0 and 1. 0 means not important, and 1 means important. You also pass the hidden state and current input into the tanh function to squish values between -1 and 1 to help regulate the network.

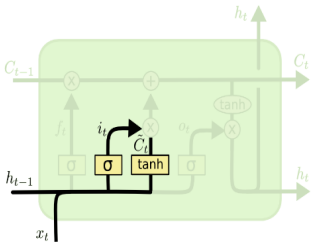
Then you multiply the tanh output with the sigmoid output. The sigmoid output will decide which information is important to keep from the tanh output.



$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \; + \; b_f\right)$$

Fig. 9. Input Gate

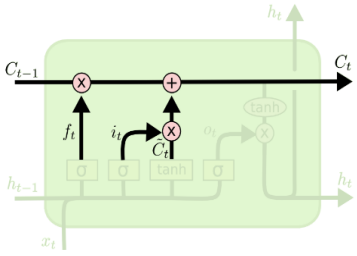
4) Forget Gate: This gate decides what information should be thrown away or kept. Information from the previous hidden state and information from the current input is passed through the sigmoid function. Values come out between 0 and 1. The closer to 0 means to forget, and the closer to 1 means to keep.



$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \; + \; b_i\right)$$
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \; + \; b_C)$$

Fig. 10. Forget Gate

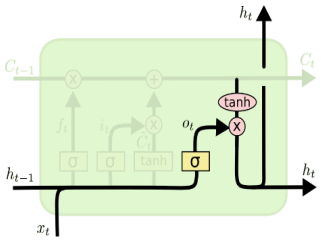
5) Cell State: Now we should have enough information to calculate the cell state. First, the cell state gets pointwise multiplied by the forget vector. This has a possibility of dropping values in the cell state if it gets multiplied by values near 0. Then we take the output from the input gate and do a pointwise addition which updates the cell state to new values that the neural network finds relevant. That gives us our new cell state.



$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

Fig. 11. Cell State

6) Output Gate: Last we have the output gate. The output gate decides what the next hidden state should be. Remember that the hidden state contains information on previous inputs. The hidden state is also used for predictions. First, we pass the previous hidden state and the current input into a sigmoid function. Then we pass the newly modified cell state to the tanh function. We multiply the tanh output with the sigmoid output to decide what information the hidden state should carry. The output is the hidden state. The new cell state and the new hidden is then carried over to the next time step.



$$o_t = \sigma\left(W_o \; [h_{t-1}, x_t] \; + \; b_o\right)$$
$$h_t = o_t * \tanh\left(C_t\right)$$

Fig. 12. Output Gate

I.e. the Forget gate decides what is relevant to keep from prior steps. The input gate decides what information is relevant to add from the current step. The output gate determines what the next hidden state should be.

7) Training: An RNN using LSTM units can be trained in a supervised fashion, on a set of training sequences, using an optimization algorithm, like gradient descent, combined with backpropagation through time to compute the gradients needed during the optimization process, in order to change each weight of the LSTM network in proportion to the derivative of the error (at the output layer of the LSTM network) with respect to corresponding weight.

A problem with using gradient descent for standard RNNs is that error gradients vanish exponentially quickly with the size of the time lag between important events. This is due to $\lim_{n \to \infty} W^n = 0$ if the spectral radius of W is smaller than 1. However, with LSTM units, when error values are back-propagated from the output layer, the error remains in the LSTM unit's cell. This "error carousel" continuously feeds error back to each of the LSTM unit's gates, until they learn to cut off the value.

C. Using LSTM to predict Stock Prices

1) The Dataset: The raw data considered was the Open, Close, High, Low and Volume Data of Stocks/Exchanges of 7 different kinds.

These 7 dataframes were merged using the common column of 'Date' and named 'common_df. A new variable called the "return" on a stock on a particular period was created and is defined as the change in price of an asset, investment, or project over that period of time, which may be represented in terms of price change or percentage change. The merged dataframe was found to have 240 entries and 43 feature variables. It was noted that all the variables in the

considered dataset are continuous in nature and have values of 'float65' datatype.

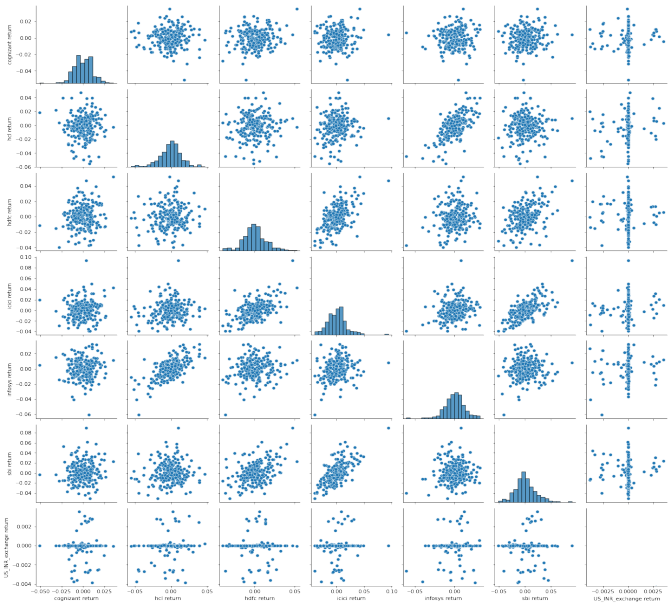


Fig. 13. Pairplot of different kinds of stocks and exhanges

2) Visualizations and insights from the analysis: The pairplots show that, the returns of HCL and Infosys show a very mild linearity. (this maight be beacuse both these companies belong to the IT and Consulting sector). It was also inferred that the returns of ICICI, HDFC and SBI show a very mild linearity. (This is due to the basic reason that all of these three companies are from the same sector i.e. financial).

On analysis, it was found that ICICI and Infosys had their worst returns on "2020-10-15". Since 2 of the companies share the same day for the worst drop, it was expected that some significant event happened that day. Which was later confirmed. ('https://www.livemint.com/market/stock-market-news/sensex-nifty-live-today-15-10-2020-nifty-nse-bse-news-updates-11602729631205.html') "Benchmark indices tumbled on global meltdown. Sensex plunged 1066 points to end at 39,728 and Nifty slipped 2.5% to 11,680. Investors lost 3,33,360.15 crore in trade on that day. Market players booked profit amid fears of a second wave of virus, as large European cities ordered clampdowns. On the Nifty, only three stocks advanced and 47 declined. The volatility index VIX rose 9% as the market breadth remained weak. All sectoral indices ended in the red with banks and IT clocking the biggest loss."

It was also infereed that Cognizant and HDFC had their best returns on "2021-02-24". Since 2 of the companies shared the same day for the highest gain, extensive analysis was done on the financial market with regards to that date and insights were found. ('https://www.nasdaq.com/articles/stock-market-news-for-feb-24-2021-2021-02-24') "Markets Bounce Back After Federal Reserve Chair Eases Concerns". It is also noteworthy that ICICI and SBI had their best returns on "2021-02-01" this was due to ('https://www.businesstoday.in/news/story/union-budget-2021-after-idfc-icici-and-idbi-conversion-into-banks-govt-proposes-new-dfi-286177-2021-02-01') "Insurance to be the game-changer with no 74% FDI, lower bad debt eases stress on banks."

Infosys's largest drop(2020-10-15) and biggest gain(2020-12-22) were considerably close to one another, so further research was done to check if anything significant happened in that time frame. It was found out that "An Interim dividend of 240% was announced on 2020-10-14. After a stock goes ex-dividend, the share price typically drops by the amount of the dividend paid to reflect the fact that new shareholders are not entitled to that payment. Dividends paid out as stock instead of cash can dilute earnings, which can also have a negative impact on share prices in the short term. The reason for this price drop is because the amount paid out no longer belongs to the company."

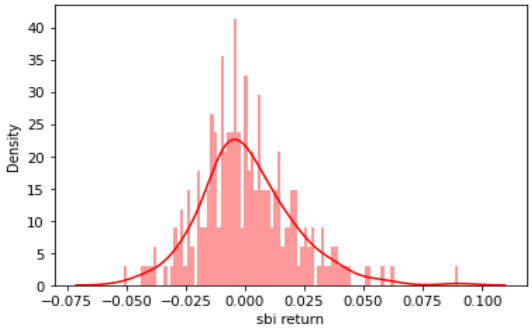


Fig. 14. Distplot of returns of SBI stocks

Taking a look at the standard deviation of the returns(Fig.14), the stocks of SBI Bank would be classified as the riskiest over the entire time period.

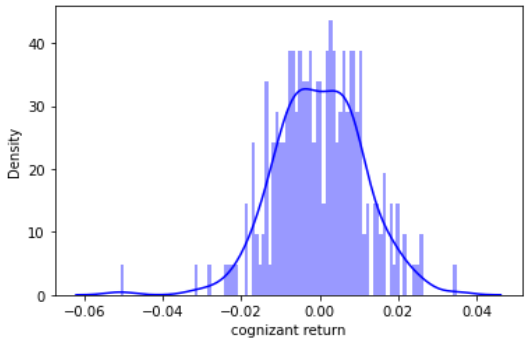


Fig. 15. Distplot of returns of Cognizant stocks

From (Fig.15.), taking a look at the standard deviation of the returns, the stocks of Cognizant would be classified as

the safest over the entire time period.

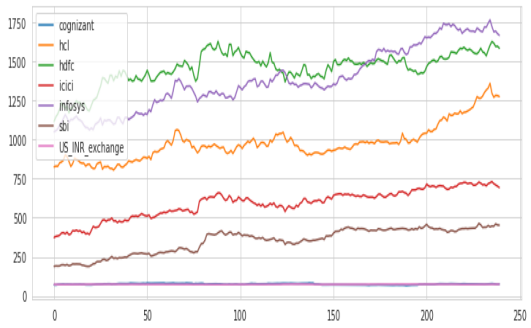


Fig. 16. Variation of the Close price of different stocks/exchanges with time

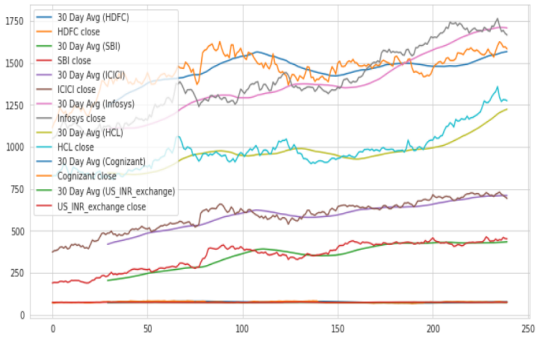


Fig. 17. Variation of the Moving Averages and close price of different stocks/exchanges with time

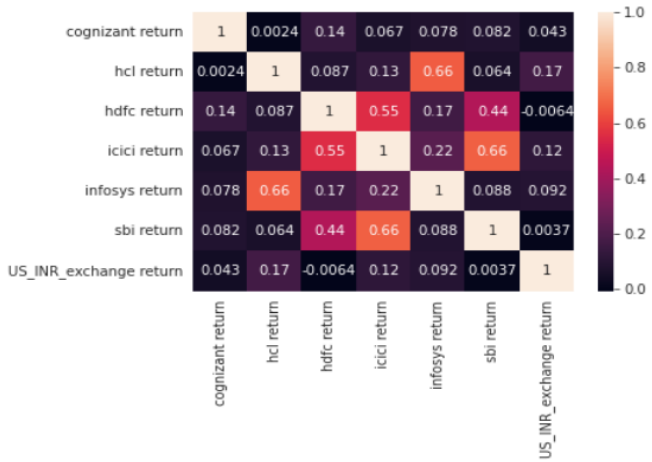


Fig. 18. Seaborn heatmap of the correlation between the Stocks Close Price

From (Fig.18.) we infer that, the Infosys and HCL returns are significantly correlated. The returns of HDFC, ICICI and SBI are also found to be mutually correlated by values close to '0.5'.

A cluster-map (Fig. 19.) was made to cluster the correlations seen in the correlation matrix heatmap together.

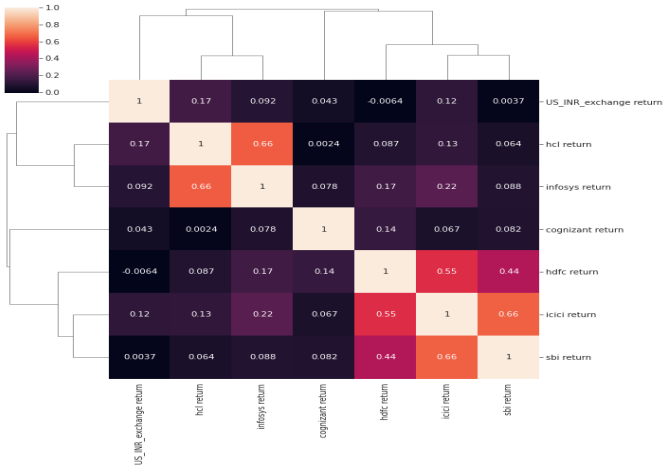


Fig. 19. Seaborn's clustermap (To cluster the correlations together)

3) Data Preprocessing: Feature scaling was performed on the data with the help of "Min-Max" Scalar from the "sklearn" library to prevent unnecessary instability in our model. The dataset was also split into the training and validation parts with the validation data being used to calculate the accuracy and performance of the model before making predictions on the test data. And the data was later converted into the most apt format to be sent into the LSTM model.

4) Building the LSTM Model: A LSTM Model has been built using the several inbuilt Dense and activation layers from the Keras library. The variables- Stock Open, High, Low and Volume were taken as the continuous inputs and were used to predict the output variable-'Stock Close' using the LSTM based model.

5) Model performance: The dataset was split into the training and validation parts and the training dataset was used to train the model to fit the given data. The training was done for 15000 epochs having 'Adam' as the model optimizer to minimize the 'mean squared error'. The decision tree model was found to fit the training data with an astounding 100% accuracy.

The performance of the LSTM model was also evaluated and, the MSE loss of the model was found to be '49.5066'. A plot of the predicted and target outputs (Fig.20.) on the validation dataset was plotted to visualize the miss-classified datapoints.

D. CONCLUSIONS

We analyzed the various aspects of the stock and exchange data provided. The variation of various stocks with respect to that of others were closely inferred using pairplots and correlation matrices and heatmaps. These correlations were later clustered using a clustermap to understand it better. Extensive analysis was done on all the stocks on how much

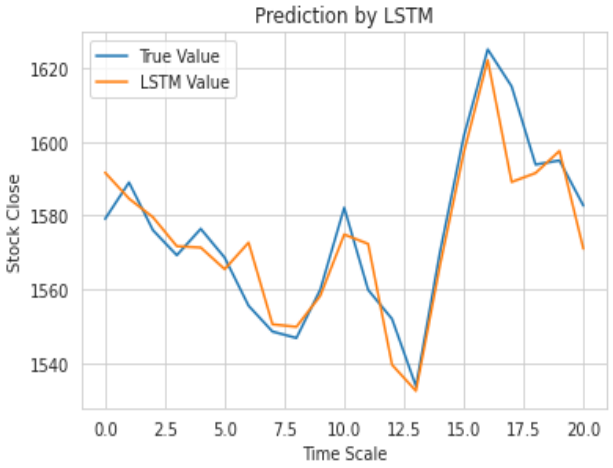


Fig. 20. The predicted and target outputs on the validation data

the highest and least returns were and when the occurred. Research was done to identify the reason for why multiple stocks saw highest rise/worst dip on same days and the facts were documented. It was also analysed and found that Infosys had seen its biggest gain and largest drop on dates considerably close to one another. Extensive research was performed on this period of time and it was found that an Interim Dividend occurred in that period which led to such a huge deviation in returns over a short period of time. The deviation of returns was also analysed for the stocks/exchanges of different companies and it was found out that the stocks of SBI bank were the riskiest over the entire time period and that of Cognizant were the safest. Plots were made to visualize different aspects of the data and moving averages were computed and graphed. A LSTM based model was build after data preprocessing and was trained on the training set of data and RMSE loss computed. The performance of the trained model on the validation data was later evaluated and,predicted and true output stock values plotted.

REFERENCES

[1] Hochreiter, Sepp Schmidhuber, Jürgen. (1997). Long Short-term Memory. Neural computation. 9. 1735-80. 10.1162/neco.1997.9.8.1735.
[2] Korstanje, Joos. (2021). LSTM RNNs. 10.1007/978-1-4842-7150-6_18.
[3] Feurer, Matthias Hutter, Frank. (2019). Hyperparameter Optimization. 10.1007/978-3-030-05318-5_1.
[4] Yang, Li Shami, Abdallah. (2020). On Hyperparameter Optimization of Machine Learning Algorithms: Theory and Practice.