2

Performance Evaluation of Machine Learning and Deep Learning Techniques: A Comparative Analysis for House Price Prediction

Sajeev Ram Arumugam[1]*, Sheela Gowr[2], Abimala[3], Balakrishna[2] and Oswalt Manoj[4]

[1]Department of AI&DS, Sri Krishna College of Engineering and Technology, Coimbatore, India
[2]Department of CSE, Vels Institute of Science, Technology & Advanced Studies, Chennai, India
[3]Department of ICE, St Joseph's College Of Engineering, Chennai, India
[4]Department of CSBS, Sri Krishna College of Engineering and Technology, Coimbatore, India

Abstract

Prediction is the act of forecasting what will happen in the future. The field of prediction is gaining more importance in almost all the fields. Machine learning techniques have been used widely for predictions also in recent time deep learning algorithms gain more importance. In this paper, we will be performing prediction over a dataset using both machine learning and deep learning techniques, and the performance of each method will be identified and compared with each other. We have used the house price dataset, which consists of 80 features, which will help to explore data visualization methods, data splitting, data normalization techniques. We have implemented five regression-based machine learning models including Simple Linear Regression, Random Forest Regression, Ada Boosting Regression, Gradient Boosting Regression, Support Vector Regression were used. Deep learning models, including artificial neural network, multi output regression, regression using Tensorflow-Keras were also used for regression.

**Corresponding author*: imsajeev@gmail.com

Rajdeep Chakraborty, Anupam Ghosh, Jyotsna Kumar Mandal and S. Balamurugan (eds.) Convergence of Deep Learning In Cyber-IoT Systems and Security, (21–66) © 2022 Scrivener Publishing LLC

The study was further extended to compare the performance of the classification models and hence six machine learning models and three deep learning models including logistic regression classifier, decision tree classifier, random forest classifier, Naïve Bayes classifier, k-nearest neighbor classifier, support vector machine classifier, feed forward neural network, recurrent neural network, LSTM recurrent neural networks were used. The models were also fine-tuned and results were also compared using performance metrics. We have split our dataset in to 70:30 ration for training and testing. In regression models random forest algorithms were performing better with MAE score 0.12, MSE score 0.55, RMSE score 0.230 and R2 score of 0.85 and in deep learning Tensorflow-Keras–based regression model was performing well with MAE score 0.12, MSE score 0.54, RMSE score 0.210 and R2-Score of 0.87, while in the other side, the classification model, random forest model, was performing good with accuracy of 89.21%, and in deep learning classification technique, feed forward neural network model, was performing good with accuracy of 89.52%. Other performance metrics including Cohen kappa score, Matthews correlation coefficient, average precision, average recall, and F1 score were also calculated to compare the performance.

Keywords: Machine learning, deep learning, KNN, SVM, CNN, RNN, prediction system, tensorflow, error detection methods, evaluation parameters, optimization techniques

2.1 Introduction

Making predictions from an existing data are always typically the toughest job, as the predictions play a major role in making decisions. With the advancement in technology, we have bunches of algorithms with us, which helps us to do the prediction easier. Machine learning algorithms gathered huge attention in doing these kinds of predictions due to its higher accuracy rates. In recent days, deep learning algorithms (a part of machine learning algorithm) are used in almost all applications for carrying out the prediction tasks.

Predictions are majorly classified into regression-based and classification-based. In regression-based algorithms, we get with a single valued output, and in classification based, we come up with 2 or more than two outputs based on the labels available in the dataset. In this work, as we try to predict the selling price of the house, these experiment falls under regression problem and hence mostly used regression algorithms are being built and their performance are measured.

The work's main goal is to figure out how well machine and deep learning algorithms function, and hence, we have to check with the

classification-based algorithms, and hence, the existing database is labeled based on the selling price, and with the modified dataset, classification algorithms are also build.

The rest of the article is organized as the part 2 discuss the related works, we have done two related researches, survey about articles which compares the performance of different ML and DL algorithms and articles that discuss about prediction of price of the house. Part 3 will be the research methodology that gives a detail of how the research work was carried on from data collection till calculating the performance of the algorithms. Part 4 describes about how the experimentation is taken place. Part 5 discussion on results, part 6 is the suggestions, and part 7 concludes the research work.

2.2 Related Research

By forecasting the price of a house based on features provided, we hope to evaluate the performance of ML and DL methods. Our primary goal is to analyze and compare the performance of the prediction system, hence the related work is narrated as two sections. In the first section of related research, we discuss about few works in which the authors tried to compare the performance of the ML and DL algorithms and in the second section we discuss about some of the works carried out for predicting the house price. Section 2.1 various related works using ML and DL algorithms are discussed and in section 2.2 works in the field of house price prediction is discussed.

2.2.1 Literature Review on Comparing the Performance of the ML/DL Algorithms

Q1 Sewak *et al* [1] presented the work of comparing the performance of machine learning algorithms for detection of malware in a system based on few features. Authors have used necessary dataset for training the system. As the dataset used had issues with data unbalancing, they have used Adaptive Synthetic (ADASYN) and have built models using machine learning-based random forest (RF) algorithm and deep learning–based deep neural network (DNN) algorithms. Various performance metrics were used for evaluating the performance and finally came up with 99.78%, 99.21% accuracy for RF and DNN methods.

Doleck *et al* [2] gave a comparative analysis report of effectiveness of online education. They have used classification-based machine and deep

learning methods over online education dataset and compared their performances. The authors have handled two dataset from MOOC and the other one from CEGEP Academic Performance. Neural networks are built using the predefined API from Keras and Tensorflow. They have also used few ML algorithms namely k-nearest neighbors (KNN), logistic regression (LR), Naive Bayes (NB), and support vector machine (SVM). In this method, they have tested the model with different optimizers, such as adadelta, adagrad, adam, adamax, nadam, and the accuracy of the models were presented using visual graphs. The accuracy using the MOOC dataset was identified to be 58.29% to 69.19% and while using CEGEP Academic Performance dataset the accuracy was identified to be 62.20% to 90.32%

Dong and Wang [3] came up with a comparative study for predicting the network intrusion. They compared deep learning approaches, such as the restricted Boltzmann machine (RBM), and back propagation (BP) with some of the traditional methods Native Bayes, random forests, decision tree, and SVM. They have used KDD-99 data set for training and testing the models which consists of data with four different classes. The dataset was holding unbalancing issue and hence synthetic minority oversampling technique (SMOTE) oversampling technique was used. Performance measures, including precision and recall, were used and concluded hybrid method combining SVM and RBM was performing better.

Liu *et al* [4] presented a comparative analysis of three deep learning methods, namely region-based convolutional neural networks (R-CNN) and expanded convolutional neural network (U-NET) in identification of birds from aerial images. Authors used Little Birds in Aerial Imagery (LBAI) dataset which consists of 34,442 photos of birds captured in a farm. The system was evaluated using some of the performance metrics including precision, recall, F1 score, and mean absolute error (MAE). U-Net model, managed to give good performance with precision of 0.861, F1 score of 0.819 and MAE of 38.5.

Chen and McKeeverSarah [5] proposed a comparative analysis for analyzing text in social media and identify the abusive contents. Authors have collected and used datasets from social media pages like Twitter, YouTube, Myspace, forum spring, Kongregate and slash dot. The goal of the research was to compare deep learning algorithms to classic machine learning algorithms, namely decision tree, logistic regression, SVM, Naïve Bayes, CNN, and RNN. For comparing the performance of the systems, they have used the metric recall. Authors concludes that SVM was performing better than other ML and DL models.

Alakus *et al* [6] considered the need of the time and proposed a model, which compares the different deep learning methods for predicting

COVID-19 infections. In the work, the authors have used 18 laboratory data consisting details of 600 patients by which the models are trained. They developed different deep learning models including artificial neural network (ANN), recurrent neural networks (RNN), long short-term memory (LSTM) and convolutional neural networks (CNN). Performance measuring metrics including precision, F1-score, recall, AUC, and accuracy score was used. They concluded the work with LSTM was performing better than other models with F1-score of 91.89%, accuracy of 86.66%, AUC of 62.50%, precision of 86.75%, and recall of 99.42%.

2.2.2 Literature Review on House Price Prediction

Ghosalkar and Dhage [7] proposed a house price prediction method using traditional linear regression model. To train and validate the model, authors have collected data for Mumbai home prices from zillo.com and magicbricks.com. Various performance parameters, such as mean absolute error (MAE), root mean squared error (RMSE), and mean squared error (MSE), were calculated and the model had a minimum prediction error of 0.3713.

Phan [8], for forecasting the price of properties in Melbourne, Australia, employed a variety of machine and deep learning approaches, such as polynomial regression, regression trees, and feed forward neural networks. Authors have used Melbourne Housing Market dataset for training and validating the models. Principal component analysis (PCA) was also used which handles the feature selection part. Finally, the mean square error (MSE) is determined to assess the system's performance, and it was identified that linear regression performed better in terms of MSE score and execution time.

Nahib *et al* [9] presented a model on predicting the real estate value of King County region in Seattle. Different prediction models like linear regression, multivariate regression models, polynomial regression were used. After evaluating the models' performance using the root mean square score, the authors came to the conclusion that none of the approaches generated appropriate models. For improving the performance of the system, authors further suggested to use a bigger dataset and also to involve a more complex model.

Varma [10] tried to predict the house price in Mumbai, they used linear regression, forest regression, boosted regression, neural networks models for predicting the price. The data analysis part and feature selection and the prediction were represented in an easily understandable manner. But the authors did not evaluate the performance and compared with other prediction models.

Madhuri *et al* [11] used datasets from Vijayawada, Andrapradesh to predict the price of homes in the city. In the study, they have implemented Ridge, Multiple linear, LASSO, gradient boosting elastic net, and Ada Boost Regression models for making the prediction. For evaluating the performance of the models, when compared to the other models employed, the authors determined that the gradient boosting approach had the best accuracy.

Panda *et al* [12] performed two predictions: to predict the salary of employees a specific number of years and predicting the real estate values. The authors have used datasets from Kaggle for training and validation. The models Simple Linear Regression (SLR) and Multiple Linear Regression (MLR) algorithms were implemented for predictions. The performance of the system was measured using R-squared value, MAE, MSE, RMSE, MDAE, and Variance Score. Finally, the authors concluded that the Multiple Linear Regression model was performing better for both the predictions when compared with other models.

Rawool *et al* [13] presented a house prediction model using linear regression, decision tree regression, K-means regression, and random forest regression. The dataset was obtained from online real estate websites and repository and have used median values to fill the empty faces in it. The root mean square error (RMSE) was used to assess the model's performance, and it was discovered that random forest models outperformed the others.

2.3 Research Methodology

The basic process of the work involves the following seven steps,

I. Data Collection
II. Data Visualization
III. Data Preparation
IV. Regression Models
V. Classification Models
VI. Performance Metrics for Regression Models
VII. Performance Metrics for Classification Models

Although house prediction falls on the regression problem, to explore about the deep learning models, we have used both regression and classification models and made predictions. the rest of the section is organized as section 3.1 describes about data collection methods, section 3.2 describes about the different data visualization techniques, Section 3.3 describes the

data preparation methodology, section 3.4 describes the working of different regression models, section 3.5 describes different classification models, section 3.6 describes different performance metrics for regression model, and finally, section 3.7 describes about different performance metrics used for evaluating the classification models.

2.3.1 Data Collection

For training and testing a machine learning or deep learning method, it is more important we hold a good dataset with all the necessary information available in it, after going through many datasets finally we end up with a public available dataset in Kaggle for house price detection [14]. The dataset holds 79 features with 2919 records in total which is split in to two csv files train and test, which holds 1460 and 1459 records, respectively. The house price of the records in the test dataset is available as a separate csv file named sample submissions.

The dataset which we collected can be used for regression methods directly as it holds the price of the house in it, but to use for classification models a categorial output is required, and hence, we have converted the price in to categorical data. The minimum price of the home is 34,900 and the max price was listed as 75500, hence we separated the price field into four classes. Class 0: houses holding the price less than 100,000, Class 1: homes priced between 100 and 200 thousand, Class 2: houses priced between 200,000 and 300,000, and Class 3: Houses which are priced more than 300,000. None of the features in the dataset was modified other than the house price. After converting the house price to class categories, we have proceeded with classification models.

2.3.2 Data Visualization

Data visualization provides us with a clear information on what is the data about and make us understand about the pattern of data available in it. Before passing the dataset to machine learning or deep learning models, it is good to have only the necessary features and remove the features, which could not help us in predictions. Data visualizations also have the following advantages, identifying patterns, better analysis, finding errors, quick action, exploring business insights, understanding the story, grasping the latest trends [15]

Visualizations, such as infographics, heatmap, fever charts, area chart, and histogram, could be used [16] and in our work we have implemented histogram, box plot, quantile plot, scatter plot, and count plot.

Histogram: a graphical bar chart that shows data on the horizontal axis and counts on the vertical axis [17]

BoxPlot: also termed as whisker plot used to represent the spread and the center of the data, this helps us to identify the min, max, mean, median, and average of the data [18]

Quantile Plot: Graphical method of identifying if two of the data are of the same distribution. It plots the again two data and explore the common distributions [19].

Scatter Plot: a sort of chart that depicts the relationship between variables by using dots to represent each variable. [20]

Count plot: a similar plot to histogram which could be used not only for numeric data, but also for categorial data [21]

2.3.3 Data Preparation

Data preparation is often referred as data preprocessing, it is the process of making modifications in the raw data before proceeding to machine and deep learning techniques. It includes handling missing records, improperly formatted, anomalies, inconsistent values, and limited attributes [22].

In our work the following data pre-processing steps are performed

a. Merging the train and test csv files
b. Identifying the null values available with the dataset and the features have more null values have to be identified and removed (using drop command)
c. Separate the dataset into numeric and categorical data
d. Identify the most prominent features in numeric data and could be converted to categorial data by converting the type from integer to string.
e. For the numeric data, we have to perform two operations (i) filling the missing values, (ii) scaling the data.
   a. kNN Imputation: a method of finding a new sample and imputing it in the place of missed values. identifies the nearest samples and takes an average of it and impute the new values [23].
   b. Standard Scalar: Scaling is a process of fitting the data in a particular scale, for which we have used standard scalar function. It scales the data to have a mean of 0(zero) and a standard deviation of 1(one). [24].

f. Converting the categorial value to numeric value using ordinal encoder. Ordinal encoding assign a unique value for every category [25].

Finally, splitting the modified data set to training and testing data set in ratio 80:20.

2.3.4 Regression Models

In our work, for estimating the price of the house, we utilized five machine learning and three deep learning models, as mentioned below.

- Simple Linear Regression
- Random Forest Regression
- Ada Boosting Regression
- Gradient Boosting Regression
- Support Vector Regression
- Artificial Neural Network
- Multi Output Regression
- Regression using Tensorflow-Keras

2.3.4.1 Simple Linear Regression

The optimum link between the input characteristics and the output parameter is determined using linear regression. (to be predicted value) in which both of them are founded to be continues [26]. The model of linear regression looks more similar to that of slope of a line and it is shown in Equation 2.1.

$$y = \alpha + \beta x \qquad \text{(Equation 2.1)}$$

Were,

y represents the y- coordinates
α represents the y intercepts
β represents the slope of the line
x represents the x- coordinates

Implementation of linear regression

i. Analyze the dataset
ii. Get the training model to build the model

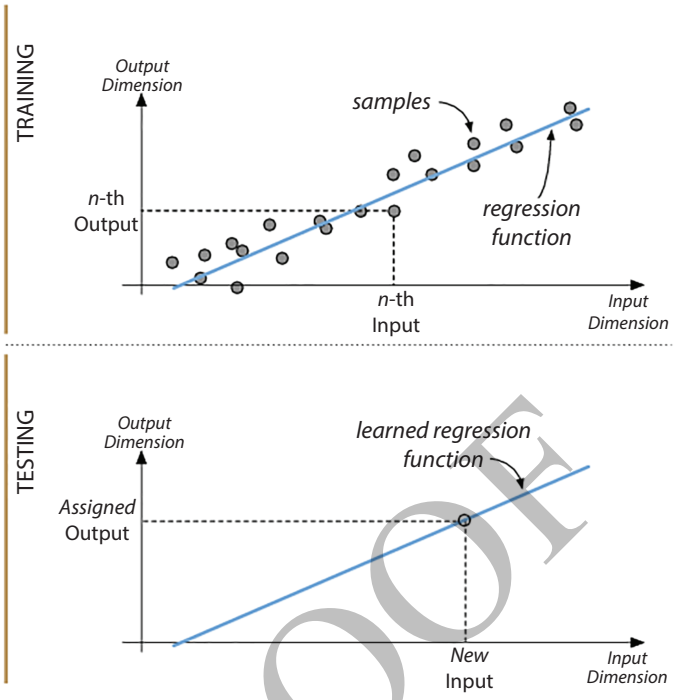

Figure 2.1 Linear Regression Classification. Source: Machine Learning of Musical Gestures [27].

iii. Using Equation 1 the model is built
iv. Using the developed model, the new predictions are made

Once the model is developed, all the necessary input parameters are fed to the model and required predictions are made, provided both the parameters and the predictors must be of continues variables, and the prediction can be clearly understood from Figure 2.1.

2.3.4.2 Random Forest Regression

Random forest method is an ensemble-based technique that can do both regression and classification problems (by creating several models and then combining all of the results). It creates multiple decision tree and make prediction for every tree and the results are finally combined for the best results [28].

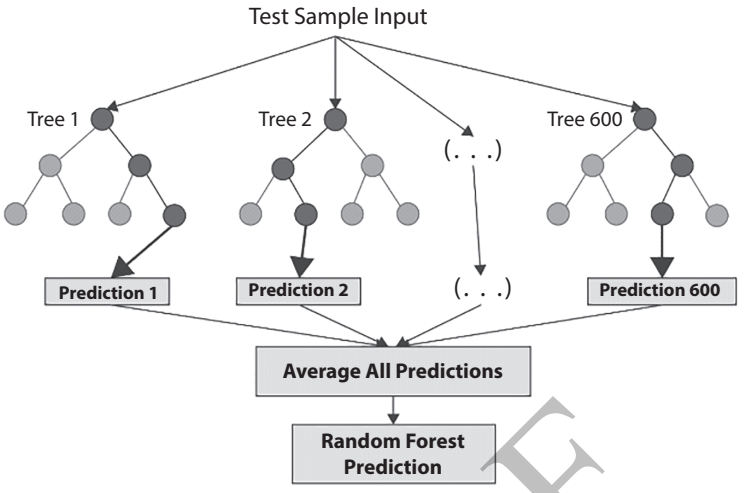

Figure 2.2 Random forest regression model. Source: A Research Paper on Loan Delinquency Prediction[29].

Implementation of forest regression

I. Get the training data from the dataset
II. Build a decision tree based in the training dataset
III. The number of trees that have to be built is decided
IV. The average of the tree's individual forecasts is used to get the final prediction

Figure 2.2 shows the working of random forest algorithm in detail.

2.3.4.3 Ada Boosting Regression

Boosting is a method of converting the weaker model into a stronger model, Ada boosting is also a type of ensemble model, it is also known as adaptive model. In this model for every instance, the weight of the model is re assigned. As it reduces the bias and the variance by boosting, it is termed as Ada Boosting. The major difference between Ada and random forest is, in random forest n number of leaves can be created from a stem, but in case of Ada only two leaves are allowed for a stem. For every stem, the performance is calculated and the performance is increased by modifying by adjusting the weights of the model [30]. The model is demonstrated in Figure 2.3.

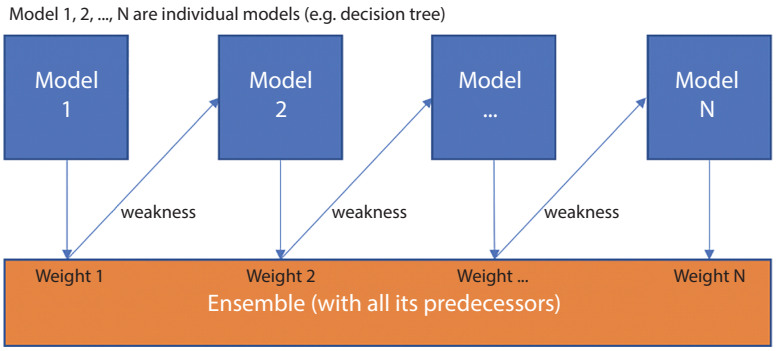

Figure 2.3 Ada boosting regression. Source: towardsdatascience.com [31].

Implementation of Ada Boosting Regression

I. Assign equal weights to all the observations
II. Classify random samples using stumps
III. Calculate Total Error of the model
IV. Calculate Performance of the Stump
V. Update Weights in the observation
VI. Update weights in iteration
VII. Final Predictions

2.3.4.4 Gradient Boosting Regression

Gradient boosting algorithm are one of the most popular regression models, also it is identified to be more effective than random forest and adaptive boosting algorithms. As like random forest and adaptive boosting, gradient boosting is also an ensemble-based model. The main ideology behind the model is to boost to the new model [32]. It continuously creates tree with boosting the previous model, and hence the new model is identified to be more superior than the previous tree as shown in Figure 2.4.

Implementation of Gradient Boosting Regression

I. From the raw datasets split the train dataset
II. Create a decision tree and fit the model using the training data
III. Calculate the error and fit the next tree by boosting the previous tree
IV. Repeat step I – iii till we get the required results.

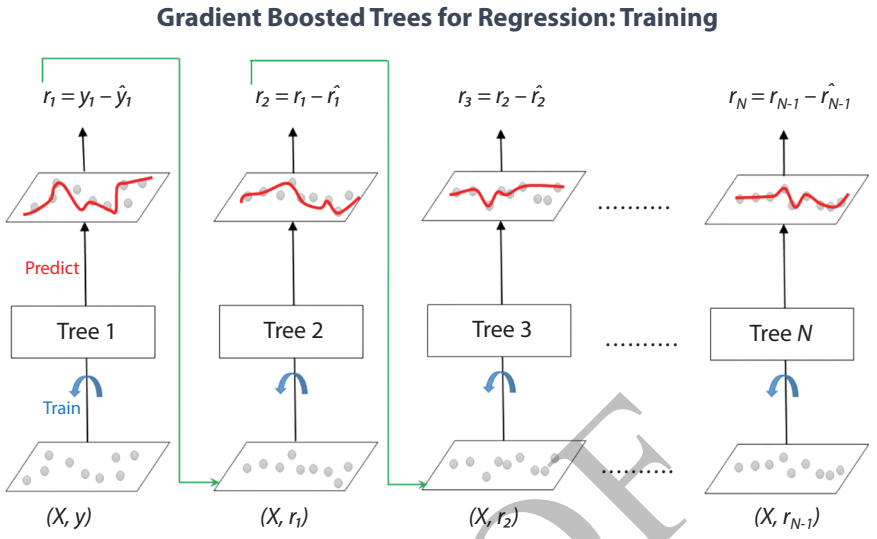

Figure 2.4 Gradient boosting regression. Source: deepnote.me [33].

2.3.4.5 *Support Vector Regression*

Support vector regression (SVM), which are considered to be more famous and most used classification-based machine learning algorithms, mostly researchers use SVM machines in classification and it is being used very rarely in Regression platforms. SVM classifier creates a hyperplane and tries to classify the data accordingly but in SVM regressor we create a decision boundary from the hyperplane on both sides termed as positive and negative hyperplane [34]. Only the points which are available inside the plane are considered and they are mapped to the hyperplane and the error rate is considered. The new predictions are made by plotting the point using the hyperplane. Figure 2.5 gives more clarity on how the hyperplane, positive and negative hyperplane are drawn and predictions are made.

The equation of the hyperplane can be written as shown in Equation 2.2

$$Y = ix + j \qquad \text{(Equation 2.2)}$$

where represents the y coordinates
j represents the y intercepts
i represents the slope of the line
x represents the x coordinates

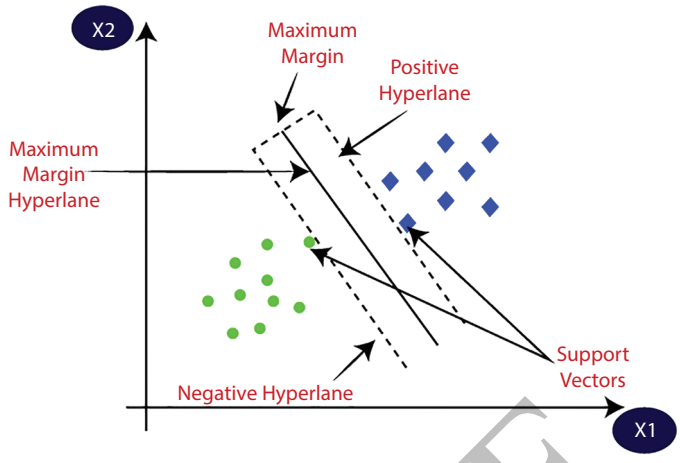

Figure 2.5 Support vector regression. Source: medium.com [35].

The equation of the positive and negative hyperplane becomes as in Equation 2.3 and Equation 2.4

$$ix + j = +m \quad \text{(Equation 2.3)}$$

$$ix + j = -m \quad \text{(Equation 2.4)}$$

And hence the hyperplane must satisfy the Equation 2.5

$$-m < Y - ix + j < +m \quad \text{(Equation 2.5)}$$

Implementation of Support Vector Regression

I. From the raw datasets split the train dataset
II. Plot the data in a spatial field
III. Identify the hyperplane
IV. Using hyperplane draw the positive and negative hyperplane

Using the hyperplane, the new coordinates are predicted.

2.3.4.6 Artificial Neural Network

Artificial neural network (ANN) will be the very first model every researcher uses to perform both regression and classification model under

deep learning methodology [36]. The basic implementation strategy of ANN is as follows

i. Reading the input data
ii. Preparing the mathematical-based prediction model
iii. Measure the error and performance of the model
iv. Making necessary changes in the model to optimise the output
v. Using the model to make predictions

Artificial neural networks have layers connected with one another; layers hold neurons which are responsible for the predictions. The neurons and layers are arranged as in Figure 2.6.

Input Layer: accepts input from the user
Hidden Layer: Performs necessary calculations for identification of the features
Output Layer: Output is presented to the user

Apart from the layers other two important factors to consider is bias and weights. The input is multiplied with weights and bias is added to it.

Every neuron except which are available with input layer produces an output based on a function called as activation function. As we are dealing

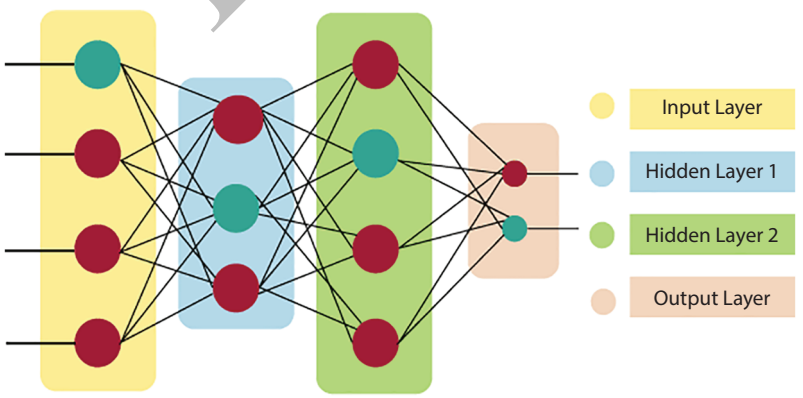

Figure 2.6 Neural network structure. Source: www.javatpoint.com [37].

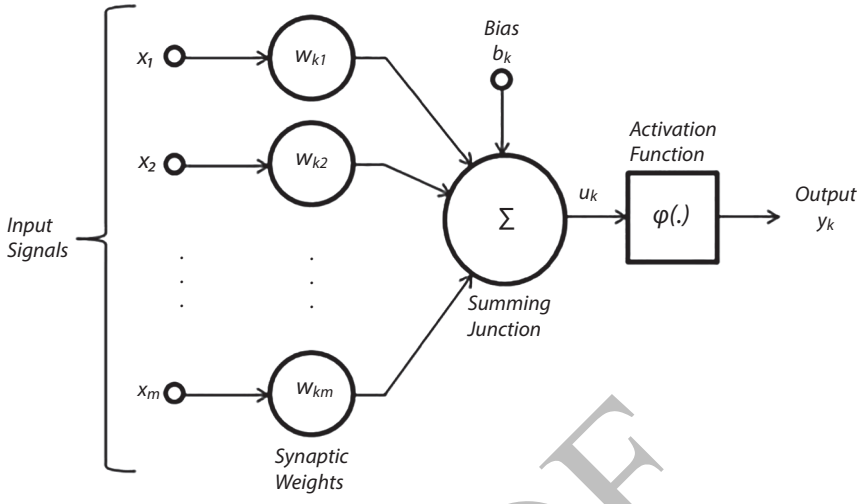

Figure 2.7 Regression function for ANN. Source: Comparison of linear regression and artificial neural network model of a diesel engine fueled with biodiesel-alcohol mixtures [38].

with regression problem simple linear regression function will be used as shown in Equation 2.6

$$Y = B + W_1x_1 + W_2x_2 + W_3x_3 + \cdots W_nx_n \quad \text{(Equation 2.6)}$$

where, Y is the variable to be predicted, B is the bias, W_1, W_2, W_3…. W_n are the weights of the attributes x_1, x_2, x_3…. x_n are the attributes.

As we are handling with regression, we need only one output, and hence, we will be having only one neuron in the output layer and the model will be looking as in Figure 2.7.

2.3.4.7 Multioutput Regression

Multi output model is a regression-based neural network model in which the output is simultaneously predicted by using the previous output back to the system. The performance of the model always depends on the quality of the output label which is predicted. And hence extra care has to be taken in labelling the output[39]. Figure 2.8 shows how the output is named in the model.

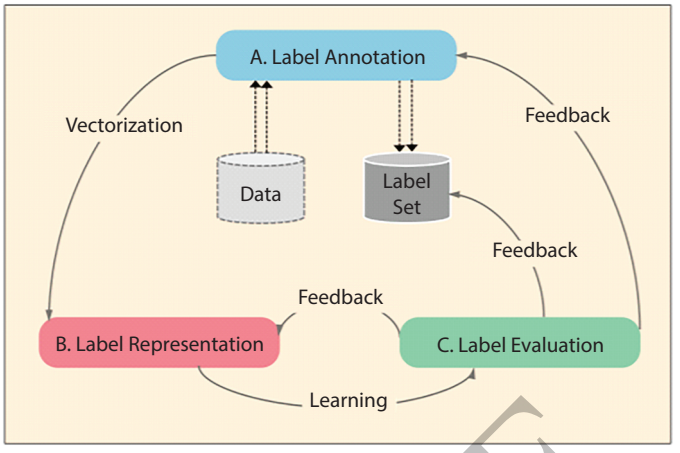

Figure 2.8 Life cycle of multioutput regression model. Source: Survey on multi-output learning [39].

Multioutput regression model is capable of handling multiple labels during regression process, which makes it unique from the other regression models beside the output model might be any type like image, text, audio or video. The multi output regression model working is displayed in Figure 2.9.

2.3.4.8 Regression Using Tensorflow—Keras

Tensorflow has an API, which is capable of performing the regression function in just three steps

i. Create the network model using Keras API belongs to Tensorflow
ii. Train the model
iii. Test the model

During training the model, it first assigns random weights to each neuron of the model, and based on the model backpropagation is applied to modify the weights to get the desired outputs. The training is done for certain iteration until we get the desired output, also we should take care such that the model does not under train or over train [41]. Once the training and testing are done, predictions can be made.

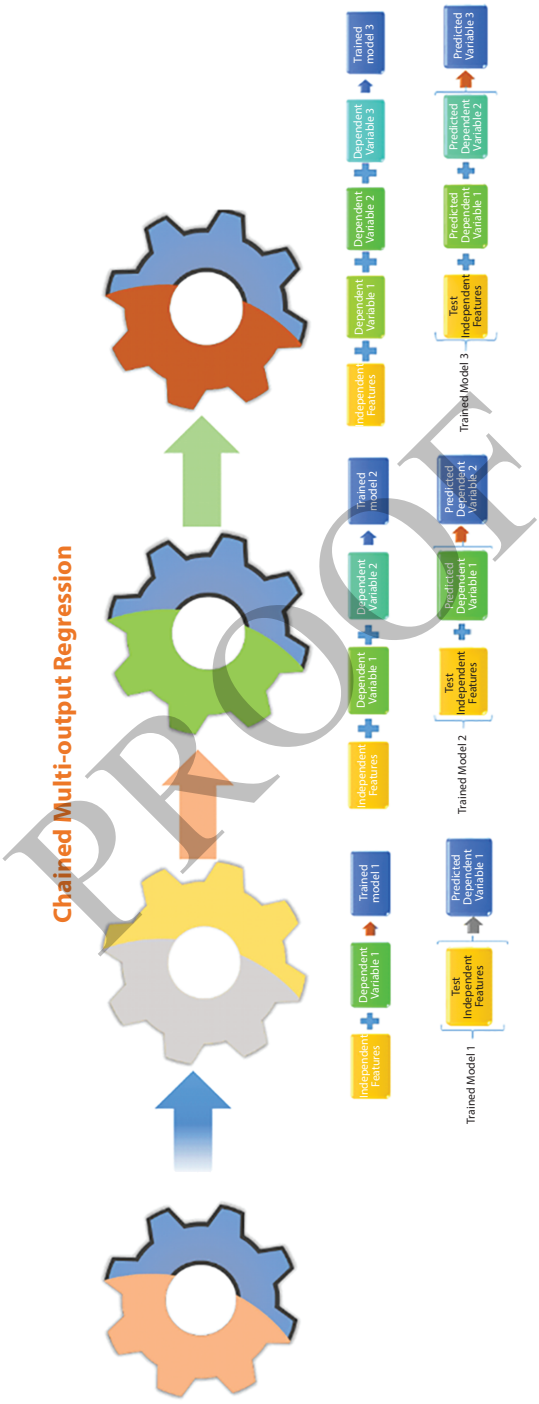

Figure 2.9 Working of multioutput regression. Source: towardsdatascience.com [40].

2.3.5 Classification Models

We have similarly carried out classification of house prices too, for that we have used six machine learning and three deep learning classification models as follows

- Logistic Regression Classifier
- Decision Tree Classifier
- Random forest Classifier
- Naïve Bayes Classifier
- k-Nearest Neighbor Classifier
- Support Vector Machine Classifier
- Feed Forward Neural Network
- Recurrent neural network
- LSTM Recurrent Neural Networks

2.3.5.1 Logistic Regression Classifier

Logistic regression is a machine learning algorithm for performing classification tasks, this algorithm is meant specially for binary classification but could be used for multi classification also. Logistic regression and logistic classification are more similar to each other, Linear regression deals with the regression problems and the other deals with the classification tasks. Among different classification algorithms, it is considered as one of the significant models as it can handle both continues as well as discrete datasets [42].

Logistic regression has an S-shaped curve between 0 and 1, for predicting a value the number is mapped in the curve and if the point lies in the above region over threshold, it is marked as positive and if on the lower side it marked as negative as shown in Figure 2.10. Logistic regression is of three types, namely binomial, multinominal, and ordinal classification.

2.3.5.2 Decision Tree Classifier

Decision tree algorithms is a machine learning algorithm which deals both the regression and classification problems. Decision tree creates a tree structure and have three types of nodes with it, namely root node, decision node, and termination node. Root node are the parent nodes and further divided to further nodes [44]. The decision nodes are nodes which are not a root node but further divides to separable nodes and termination nodes

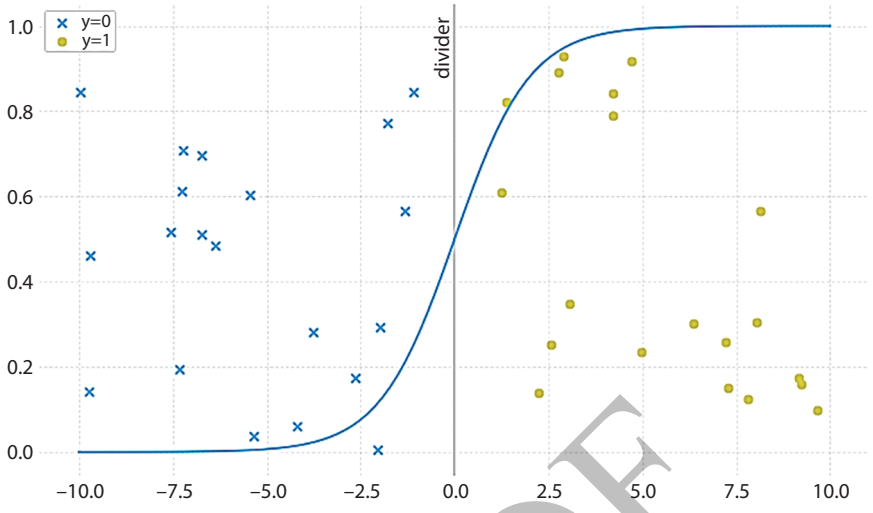

Figure 2.10 Logistic regression. Source: towardsdatascience.com [43].

does not have any other branches and it is the last node in the branch as shown in Figure 2.11.

Decision tree uses multiple algorithms to create a split in the node and creating the next nodes which could be terminal or decision node. Identifying the root node is a complex task in decision tree, one possible way for identifying the root node might be random selection but the results might not be as expected and reduces the accuracy of the system,

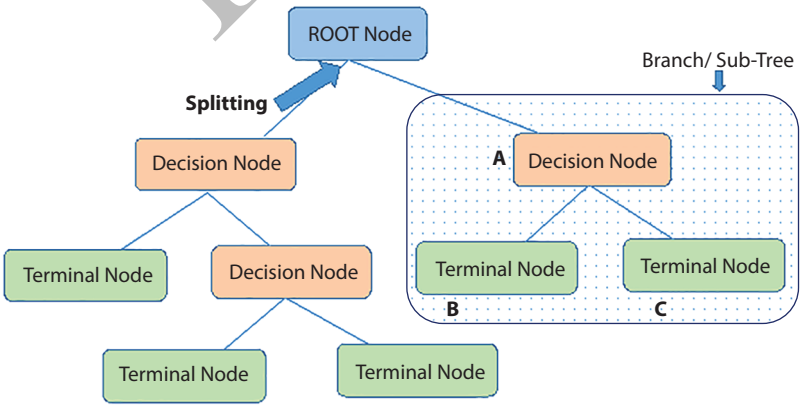

Figure 2.11 Decision tree classifier. Source: kdnuggets.com [45].

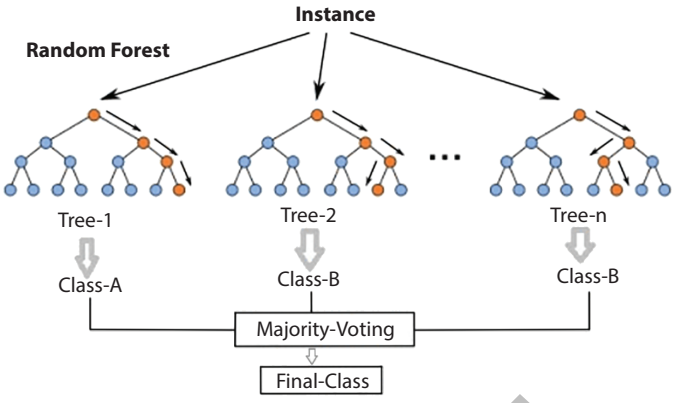

Figure 2.12 Structure of random forest classification. Source: medium.com [47].

hence some of the criteria like Gini Index, Gain Ratio, or Chi Square could be used for deciding the Root node.

2.3.5.3 Random Forest Classifier

Random forest classification task is already described in section 3.4.2, the model described in 3.4.2 was a regression algorithm which takes the average of the score and predict the values [46]. In the case of classification task and voting is done for all the trees and the output, which have the highest vote is chosen as the final output of the model as shown in Figure 2.12.

2.3.5.4 Naïve Bayes Classifier

Naïve bayes classifier is one of the Bayes theorem-based classification algorithm, which makes prediction on the basis of probabilities, and is applied in most of the classification applications including, sentiment analysis, spam detection in mails, etc [48]. Bayes theorem is mathematically described as shown in Equation 2.7

$$P(A \mid B) = \frac{P(B|A)P(A)}{P(B)} \qquad \text{(Equation 2.7)}$$

where P(A|B) chance of A occurring in event B; P(B|A) change of B being true; P(A) chances of A occurring; P(B) chances of B occurring.

Implementing Naïve Bayes

i. Calculate the prior probabilities
ii. identify the likelihood probabilities
iii. substitute all the detected values in the bayes equation and predict the class.

2.3.5.5 K-Nearest Neighbors Classifier

Because of its benefits, such as prediction power and classification speed, the K-nearest neighbour method is one of the most widely used machine learning algorithms in classification tasks. It can also be used for regression problems but not frequently used because of its adaptiveness toward the regression problems. Basically K-nn finds the similarity of new cases and the existing cases and put in a class which have maximum matches. Unlike other algorithms it does not get trained completely using the dataset, instead it stores the dataset and when new variables approach, it performs calculation on the dataset and make predictions [49]. The most important factor that has to considered in K-nn is the identifying the nearest points in the algorithm. The classification is carried out as shown in Figure 2.13.

Implementing K-nn Algorithm

i. Identifying the number of neighbours (K value)
ii. Calculate the distance between the new variable and the K-points using Euclidean distance
iii. Count the number of classes in the k-points
iv. Assign the new variable to the class which holds maximum k-points

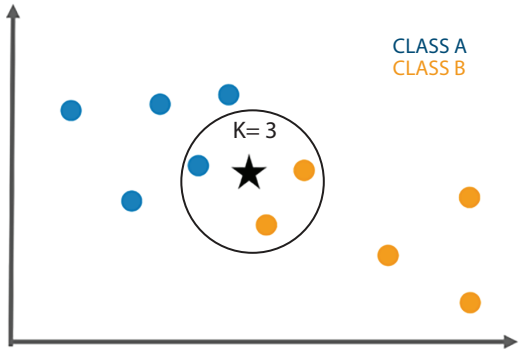

Figure 2.13 Classification based on K-nn algorithm. Source: edureka.co [50].

2.3.5.6 Support Vector Machine Classifier (SVM)

Support vector machine (SVM) algorithm which is often used in both regression and classification tasks. The algorithm is already described in section 3.4.5. the major difference is as it's a regression the score was calculated and if used for classification the class is identified using the hyper plane.

2.3.5.7 Feed Forward Neural Network

Feed forward neural network is a supervised learning deep learning machine. In this type of network, the information travels only in forward direction, i.e., from the input layer to the hidden layers, and then from the hidden layer to the output layer as shown in Figure 2.14.

A pattern is displayed in the input layer of a feed forward neural network during the learning process, and it is transmitted through the network's subsequent levels until it reaches the output layers. The output layer's neuron count is equal to the number of class labels [52]. The model's output is compared to the real output, and if there is a significant difference, the model is retrained by changing the weight of the neurons. During the classification phase the weights will be fixed and they are not adjusted, the output class is decided by the pattern mapping with the existing patterns.

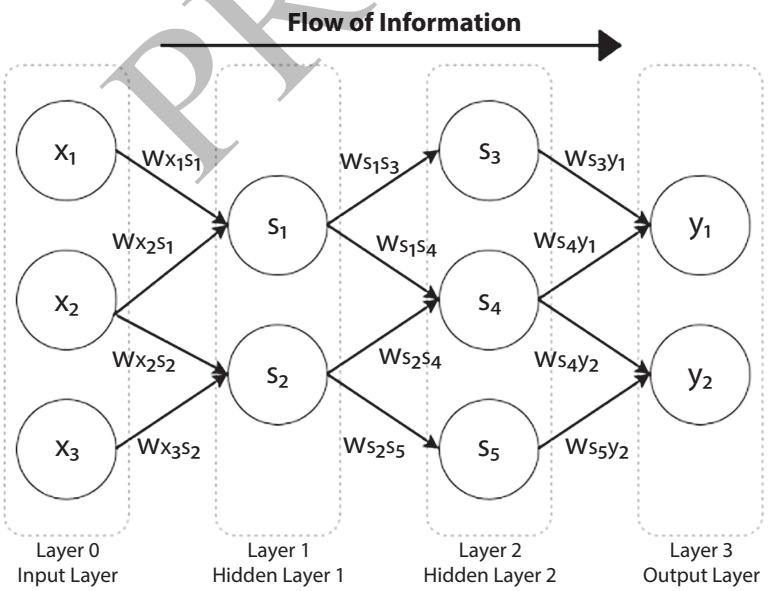

Figure 2.14 Feed forward network. Source: brilliant.org [51].

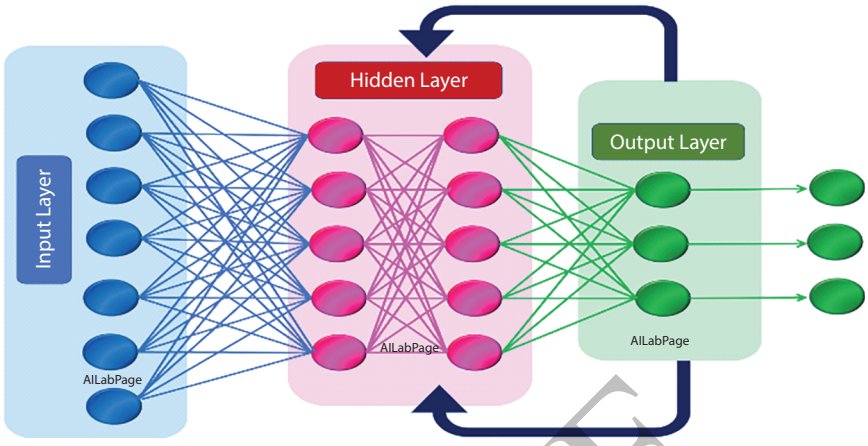

Figure 2.15 Recurrent neural network architecture. Source: medium.com [54].

2.3.5.8 *Recurrent Neural Networks*

Recurrent neural network (RNN) are network-based models used commonly for sequential data. It holds an input memory which helps us to save the previous outputs, which helps in classification of the sequential data [53]. In feed forward network, the information passes from input to output and never comes back, whereas in recurrent neural network the information cycles through a loop as in Figure 2.15.

RNN varies from other networks in that it contains two inputs: as other networks, the current input and the recent variable's output. This model uses backpropagation for optimising the output. It also uses gradient descent which reduces the function.

2.3.5.9 *LSTM Recurrent Neural Networks*

LSTM neural networks, are specialised network to solve pattern-based predictions. This LSTM model, like the Feed forward model, has input, hidden, and output layers. LSTM layer consists of self-connected recurrent blocks, called as memory blocks. Each block consists of one or more recurrent networks connected to it as shown in Figure 2.16.

The three gates of LSTM are input, forget and output gate, all these gates are fully dependent on the previous hidden layer. The output of the network is decided by the current cell state. These cell states are transferred from one cell to another by the tanh functions.

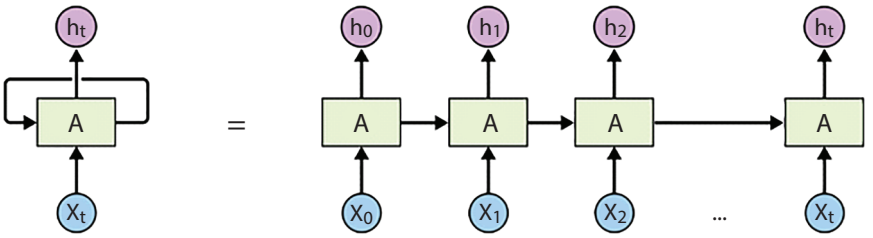

Figure 2.16 Architecture of LSTM network model. Source: pydeeplearning.weebly.com [55].

2.3.6 Performance Metrics for Regression Models

Performance metrics are used to identify how efficient our model is prediction. There are many performance measures in which we are using few of the metrics, namely mean absolute error (MAE), mean square error (MSE), root mean square error (RMSE), and R-Squared value (R2 score) is used in this work.

Mean Absolute Error (MAE)
It is used to calculate the average magnitude of prediction mistakes. The MAE is defined in Equation 2.8. A perfect model is it have an MAE score of 0.0.

$$MAE = \frac{\sum_{i=1}^{n} | y_i - x_i |}{n} \qquad \text{(Equation 2.8)}$$

where y_i is the predicted values, x_i is the true value, n is the data count

Mean Square Error (MSE)
The mean of the square difference between the real and forecasted values is calculated using MSE as shown in Equation 2.9. A good model will be holding a MSE value of 0.0

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \hat{y}_i \right)^2 \qquad \text{(Equation 2.9)}$$

where $\left(y_i - \hat{y}_i \right)$ is the difference between the expected and real values, n is the data count.

Root Mean Square Error
Root mean square error (RMSE) square root of the mean square error value, it tells us how efficiently the data is fit in the model and also how close the predictions of the model is made and is expressed as in Equation 2.10.

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{n} \left(y_i - \hat{y}_i\right)^2}{2}} \qquad \text{(Equation 2.10)}$$

R-Squared value
R-Squared value (R2 score) is closely related to MSE but not the same. It denotes the variance of dependent variable (target variable) from the independent variable (features). It is calculated using Equation *2.*11. The model is identified to be more efficient when the score is 1.0

$$R^2 = 1 - \frac{\text{Unexplained Variation}}{\text{Total Variation}} \qquad \text{(Equation 2.11)}$$

2.3.7 Performance Metrics for Classification Models

The dataset which is used here has four classes in the target variable and hence multiclassification models will be used and the performance metrics including Accuracy, Cohen kappa Score, Matthews correlation coefficient, Precision, Recall, and F1-Score are used.

Accuracy
The ratio of correct predictions to the total number of values in the dataset is called accuracy. Accuracy score will be also depending on the distribution of classes in the dataset. If the distribution of the class is not even, it might result in drop in accuracy.

Cohen Kappa Score
It measures how closely the prediction is made. It can be calculated using two scores total accuracy and random accuracy and is calculated using the Equation 2.12

$$kappa = \frac{(Total\ Accuracy - Random\ Accuracy)}{(1 - Random\ Accuracy)} \qquad \text{(Equation 2.12)}$$

Matthew Correlation Coefficient (MCC)
Matthew Correlation Coefficient is the measure of quality of classification system. MCC can be calculated Equation 2.13

$$MCC = \sqrt{PPV * TPR * TNR * NPV} - \sqrt{FDR * FNR * FPR * FOR}$$

(Equation 2.13)

where PPV: Positive Predictive Value, TPR: True Positive Rate, TNR: True Negative Rate, NPV: Negative Predictive Value, FDR: False Discovery Rate, FNR: False Negative Rate, FPR: False Positive Rate, and FOR: False Omission Rate.

Precision and Recall
Precision is defined as the proportion of true positives to all positives, while recall is defined as the model's ability to accurately identify true positives. As the model is multi classification the precision and recall are calculated for each class and averaged to get the precision and recall score.

F1 Score
F1 is the measure of test accuracy and using precision and recall are calculated by applying those values in Equation 2.14

$$F1\ score = 2\frac{(Precision * Recall)}{(Precision + Recall} \qquad \text{(Equation 2.14)}$$

2.4 Experimentation

The entire implementation was carried over in Google colab platform using python language.

The experimentation process is as follows

- Download the required dataset and upload to the colab platform or google drive for using it
- Necessary library files are installed/imported to the environment
- Data visualization is performed and data is pre-processed
- For regression and classification, the appropriate machine learning and deep learning models are created, trained, and evaluated.

- Necessary validation metrics are calculated and results are tabulated and compares using graphs.

2.5 Results and Discussion

The first step of the work will be exploring the dataset. Table 2.1 describes the dataset and its variables. The dataset consists of 79 variables, every variable is not necessarily required for the predictions however for training purpose all the variables have been included. Most of the variables are numerical and few are categorial data types.

The dataset holds 80 features and have to verify the missed data percentage and identified that five variables hold highest missing percentage: PoolQC-99.52%,MiscFeature-96.30%, Alley-93.76%, Fence-80.75%, FireplaceQu-47.26%. these variables holding highest missing values will be deleted in the data pre-processing step. Few other data variables also holds some missing values but as those variables could be important in predicting the target variables, we are not neglecting it. One of such variables is LotFrontage which holds 17% missing values, but as it could be important variable in detection of price of the house we are not including it in the remove list.

The target value for the dataset is house sale price, and hence we try to visualize the variable before and after logarithm transferred. For this box plot and histogram plots from matplot library is used and shown in Figure 2.17. From the plots its clear that the price of the house is distributed widely between 34,900 and 755,000 USD. Histogram plots the amount of data in the price range and box plot shows the lower and highest value also the mid value of the variable also could be identified.

The dataset holds both the numeric and categoric data, as on numeric data it can be directly fed for training but considering categorial data it is not the case, hence the categorial variables are visualized to identify the distribution of the data, to have an understanding of how to visualize the categorial data, we have done few visualizations for the variables SaleCondition, and OverallQual. In which the first two are plotted using Histogram which is shown in Figure 2.18.

From the histogram few things will be clear, the number of homes is too high if the sale condition is normal and also the number of homes sold is high for the overall quality being 5 to 8. Such interpretations can be made by visualizing these variables. Similarly, box plot also could be used to visualize the variables, one such is visualizing the variable neighbourhood which holds 25 categories in it and shown in Figure 2.19. We could

Table 2.1 Data description in the Dataset.

Variable Name	Description	Variable Name	Description
MSSubClass	Type of property involved in sale.	BsmtCond	Basement condition score
MSZoning	Tells the zone of the house	BsmtExposure	Basement Exposure (type of wall)
LotFrontage	Property front facing street type	BsmtFinType1	finished area in the basement
LotArea	Size of the house in square feet	BsmtFinSF1	square feet of the home after completion
Street	The street which takes us to the property	BsmtFinType2	finished area in the basement (if multiple types)
Alley	property access from the alley	BsmtFinSF2	finished area of property in sq. feet—type 2
LotShape	Shape of the saleable property	BsmtUnfSF	Unfinished area of basement measured in sq.feet
LandContour	Flat portion available in the property	TotalBsmtSF	Total available area in the basement.
Utilities	Utilities available if any	Heating	Available heaters
LotConfig	Configuration of Lot	HeatingQC	QC of the heater
LandSlope	Slope of the property if available	CentralAir	Is air conditioning available?
Neighborhood	Nearest landmarks	Electrical	Electrical system available in the home
Condition1	Proximity conditions	1stFlrSF	Area of first floor in Sq.feet

(*Continued*)

Table 2.1 Data description in the Dataset. (*Continued*)

Variable Name	Description	Variable Name	Description
Condition2	Proximity conditions available incase of more floor	2ndFlrSF	Area of second floor in Sq.feet
BldgType	Type of the Dwelling	LowQualFinSF	Finished square feet
HouseStyle	Style of the Dwelling	GrLivArea	Sq. feet of living area
OverallQual	Overall Quality of house	BsmtFullBath	No of bathrooms in basement(full)
OverallCond	Overall Condition of house	BsmtHalfBath	No of bathrooms in basement(half)
YearBuilt	Year of construction	FullBath	No of bathrooms (full)
YearRemodAdd	Year of remodel(if any)	HalfBath	No of bathrooms (half)
RoofStyle	Roof Type	Bedroom	No of bedrooms above grade
RoofMatl	Material used for Roof	Kitchen	No of kitchens above grade
Exterior1st	Exterior covering	KitchenQual	Quality of kitchen
Exterior2nd	Second Exterior covering if available	TotRmsAbvGrd	No of rooms over grade point
MasVnrType	Type of Masory vener	Functional	Home functionality
MasVnrArea	Masonry veneer area in sq. feet	Fireplaces	Availability of fireplaces

(*Continued*)

Table 2.1 Data description in the Dataset. (*Continued*)

Variable Name	Description	Variable Name	Description
ExterQual	Exterior quality of the house	FireplaceQu	Fireplace quality of the house
ExterCond	Exterior condition	PavedDrive	Paved driveway
Foundation	Type of the Foundation	WoodDeckSF	Wood deck area
BsmtQual	Basement Height		
GarageType	Location of the Garage	GarageCars	No of car parking avilable
GarageYrBlt	Year of the Garage construction	GarageArea	Area of garage
GarageFinish	Interior structure of the garage	GarageQual	Garage quality point
OpenPorchSF	Open porch available area	GarageCond	Present condition of Garage
EnclosedPorch	Enclosed area	PoolArea	If pool available, size of the pool
3SsnPorch	porch area of the seasons	PoolQC	Quality of the pool in the house
ScreenPorch	Screen size of the porch	Fence	Quality of the fense in the house
YrSold	Year Sold (YYYY)	MiscFeature	Miscellaneous features
SaleType	Type of sales	MiscVal	Net worth of miscellaneous feature available in the house
SaleCondition	Condition of sale	MoSold	Month of the house being sold

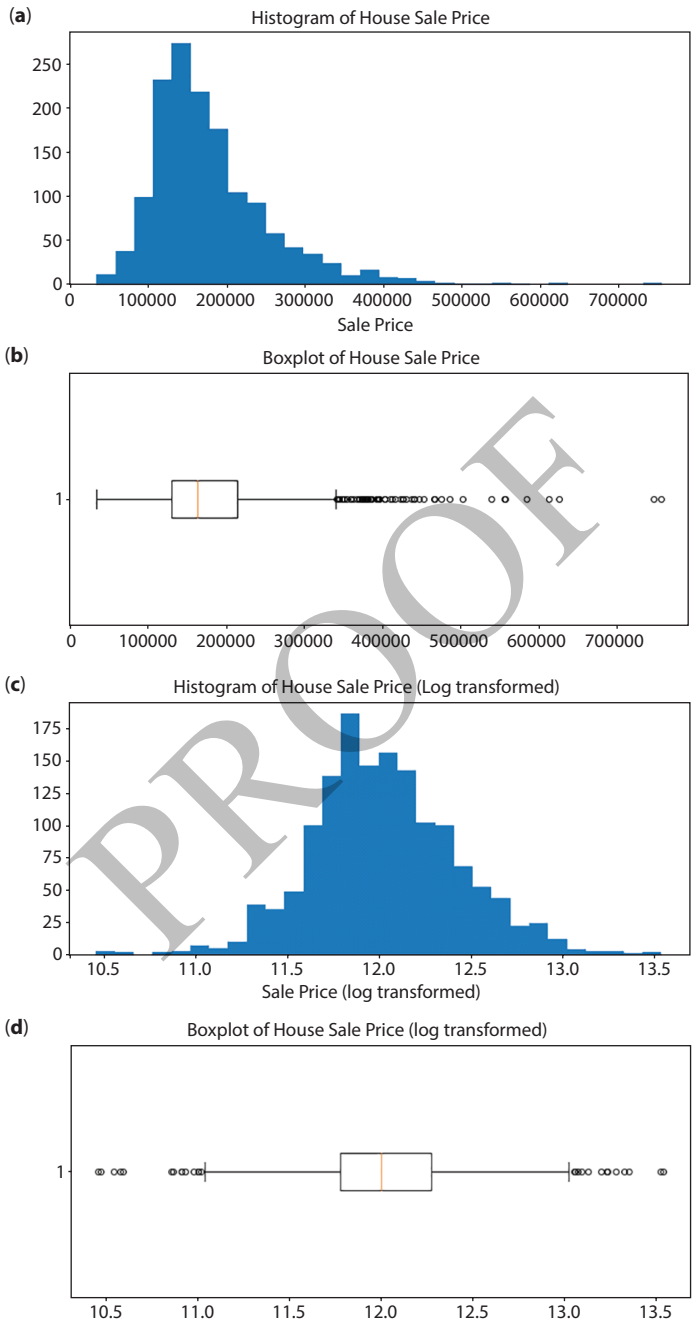

Figure 2.17 Histogram plot (a)(c) and Box Plot (b)(d) of the target variable.

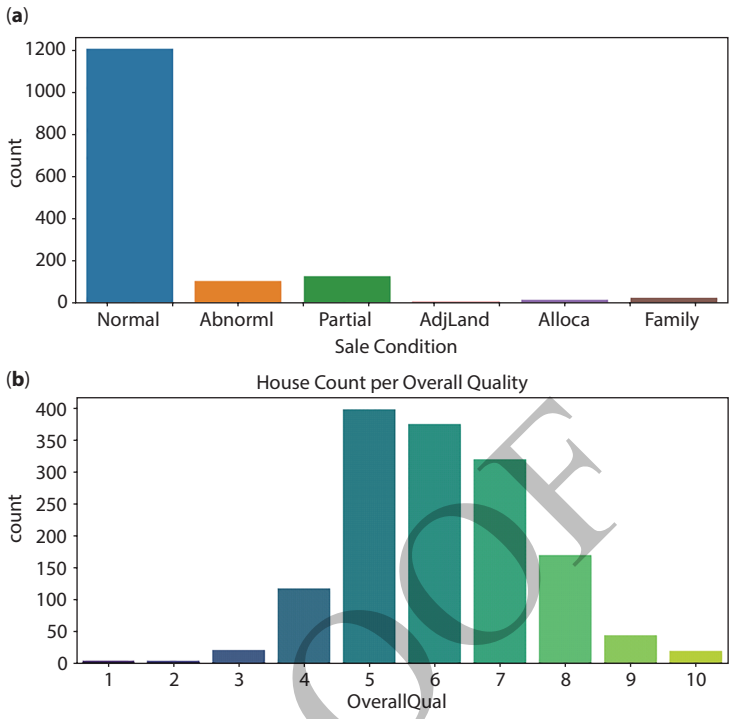

Figure 2.18 Histogram plot for the categoric variable (a) SaleCondition and (b) OverallQual.

identify the distribution of the category on sale of the home and categories NridgHt and StoneBr are having high count and this variable could be a strong predicted due to its distribution.

The next step will be data preparation, we have already listed the top 5 features which have more missing variables and hence the variables PoolQC, MiscFeature, Alley, Fence, and FireplaceQu is removed from the dataset. After removing the 5 variables we will be holding 75 variables with us, in which 39 is categorial and 36 numerical variables.

Few of the variables, including MSSubClass, OverallQual, OverallCond, YearBuilt, YearRemodAdd, GarageYrBlt, MoSold, YrSold, are strong predictors with numeric data in it, hence we change these eight variables into categorical data by converting the type of the variable from integer to text making the number of categorial data to 47 and numeric data to 28.

The most important task with the categorial data is to converting it to numeric data for which one hot encoder is used. The data before and after applying one hot encoder is shown in Figure 2.20.

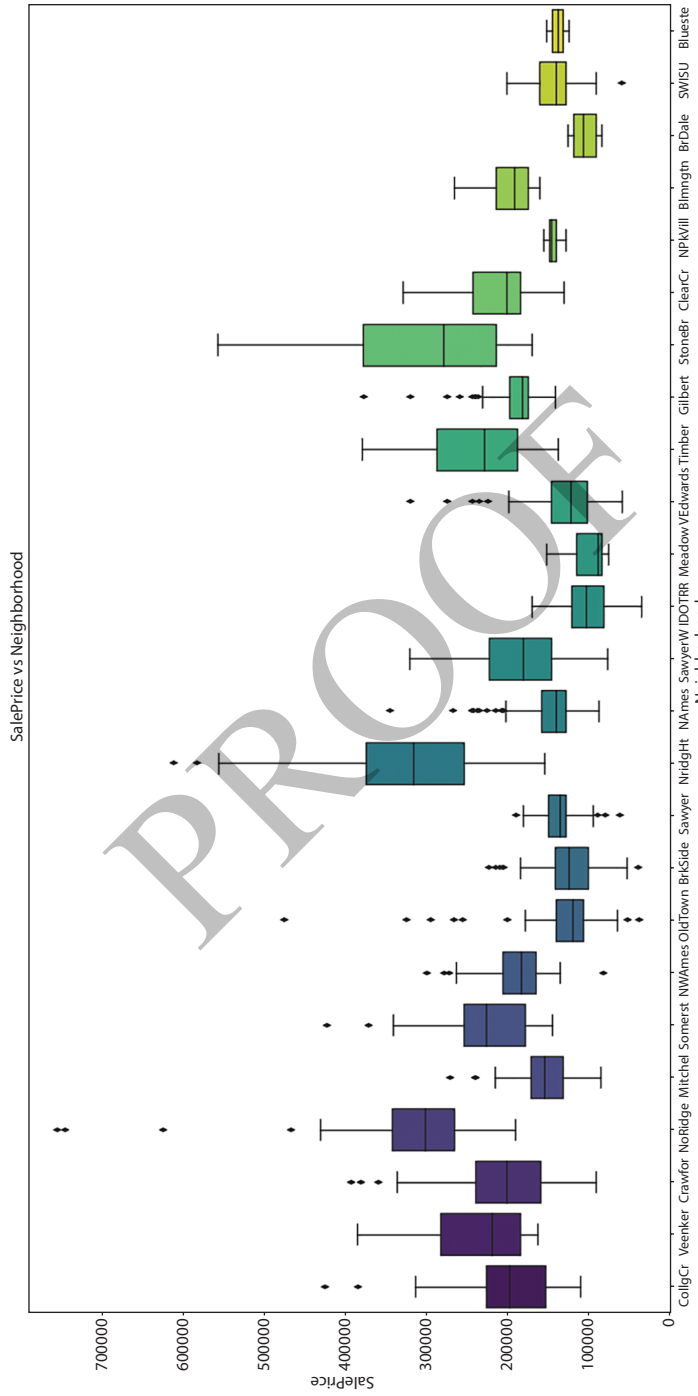

Figure 2.19 Distribution of categories in neighborhood variable.

MSSubClass	MSZoning	Street	LotShape	LandContour	Utilities	LotConfig
120	RL	Pave	Reg	Lvl	AllPub	Inside
60	RL	Pave	Reg	Lvl	AllPub	Inside
20	RL	Pave	Reg	Lvl	AllPub	Inside
75	RM	Pave	IR1	Bnk	AllPub	Corner
120	RL	Pave	Reg	Lvl	AllPub	Inside

(**a**) Before One hot encoder

MSSubClass	MSZoning	Street	LotShape	LandContour	Utilities	LotConfig	L
0.0	3.0	1.0	3.0	3.0	0.0	4.0	
9.0	3.0	1.0	3.0	3.0	0.0	4.0	
4.0	3.0	1.0	3.0	3.0	0.0	4.0	
11.0	4.0	1.0	0.0	0.0	0.0	0.0	
0.0	3.0	1.0	3.0	3.0	0.0	4.0	

(**b**) After One hot encoder

Figure 2.20 Categorial variables before and after one hot encoder.

After applying one hot encoder and converting into numeric, still we have few missing variables and hence KNN Impuser is used. The categorial data is now ready, on the other side the numerical data have to be prepared for which the data are applied to KNN Impuser and standard scalar for filling the missing values and scale the data. Finally, the numeric and categorial data are combined together.

With the processed data, all regression models were built, trained, and tested, and the performance metrics for machine learning systems were tallied in Table 2.2 and could be identified that random forest regression performs good than other models with R2score of 0.85.

The deep learning methods were also implemented, and the first model used was artificial neural network-based regression model, and the summary of the model is shown in Figure 2.21. In the ANN based model, we have used input layers with 90 neurons and capable of getting 75 features as input and activation function was Relu. Following the input layer, two hidden layers with 90 neurons each were added and the Relu activation function was employed, followed by the output layer with one neuron and the linear activation function. The loss function is mean squared logarithmic error, the optimizer is ADAM, and the performance metrics are MSE.

The ANN-based model was trained for 100 Epochs with validation split of 20% and batch size of 50. The results of the model were compromising with lower MSE value of 0.36. In multi output regression the layer count was similar to the previous model with one input, two hidden and

Table 2.2 Performance metrics for regression-based machine learning models.

	MAE	MSE	RMSE	R2-Score
Linear Regression	0.30	0.18	0.043	0.50
Random Forest Regression	0.12	0.55	0.230	0.85
Gradient Boosting Regressor	0.14	0.6	0.240	0.83
Ada Boost Regressor	0.28	0.14	0.037	0.62
Support Vector Regressor	0.37	0.37	0.061	0.57

one output, he_uniform initializer is used . SGD optimizer, mean squared logarithmic error, loss function, and MSE metrics are utilised to build the model. The model is trained with 100 Epochs with validation split of 20% and batch size of 50.

The final model in regression to be implemented is Tensorflow-Keras–based regression model. The loss function curve for the three deep learning models is shown in Figure 2.22, and the performance metrics for the models are tabulated in Table 2.3.

On the other side the Classification tasks are being carried out using both the machine and deep learning methods, the performance metrics Accuracy, Matthews Correlation coefficient, Cohen kappa Score, Average Recall, Average Precision and average F1 score are computed. For precision, recall and f1score we are considering the average as we will have values for every class individually. The precision and recall for the machine learning techniques is shown in Figure 2.23.


```
Model: "sequential_6"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_18 (Dense)             (None, 90)                6840
_________________________________________________________________
dense_19 (Dense)             (None, 90)                8190
_________________________________________________________________
dense_20 (Dense)             (None, 90)                8190
_________________________________________________________________
dense_21 (Dense)             (None, 1)                 91
=================================================================
Total params: 23,311
Trainable params: 23,311
Non-trainable params: 0
_________________________________________________________________
```

Figure 2.21 Artificial neural network model for regression.

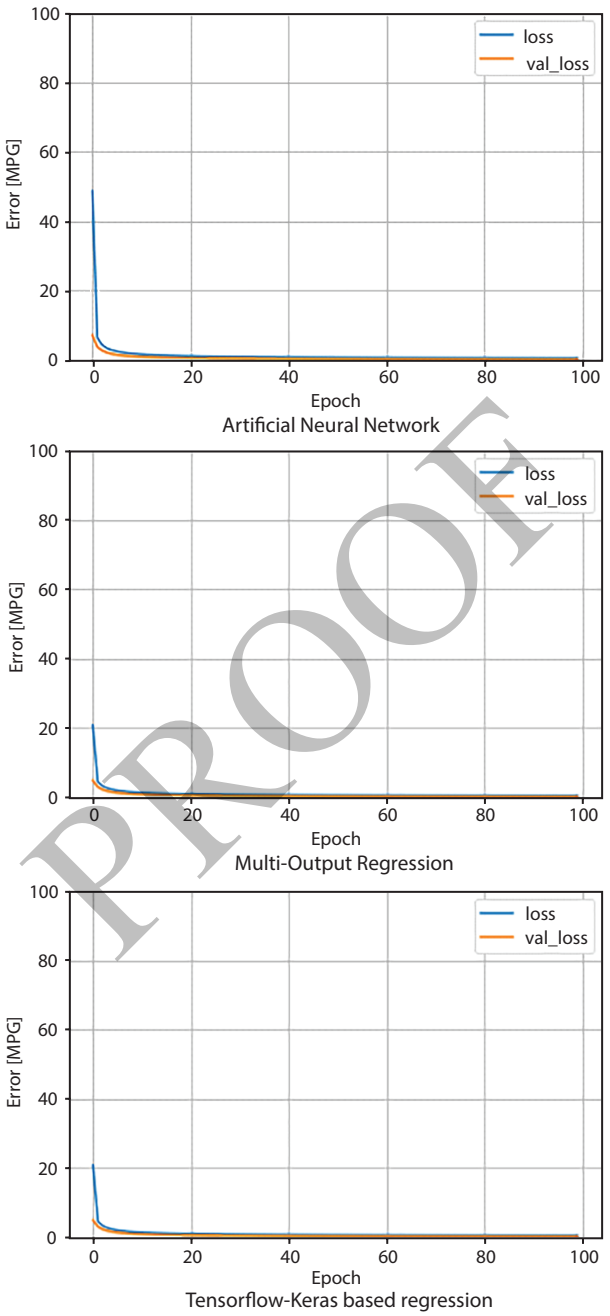

Figure 2.22 Mean_squared_logarithmic_error loss for deep learning models.

Table 2.3 Performance metrics for regression-based deep learning models.

	MAE	MSE	RMSE	R2-Score
Artificial Neural Network	0.16	0.7	0.340	0.86
Multi-Output	0.19	0.6	0.230	0.84
Tensorflow-Keras	0.12	0.54	0.210	0.87

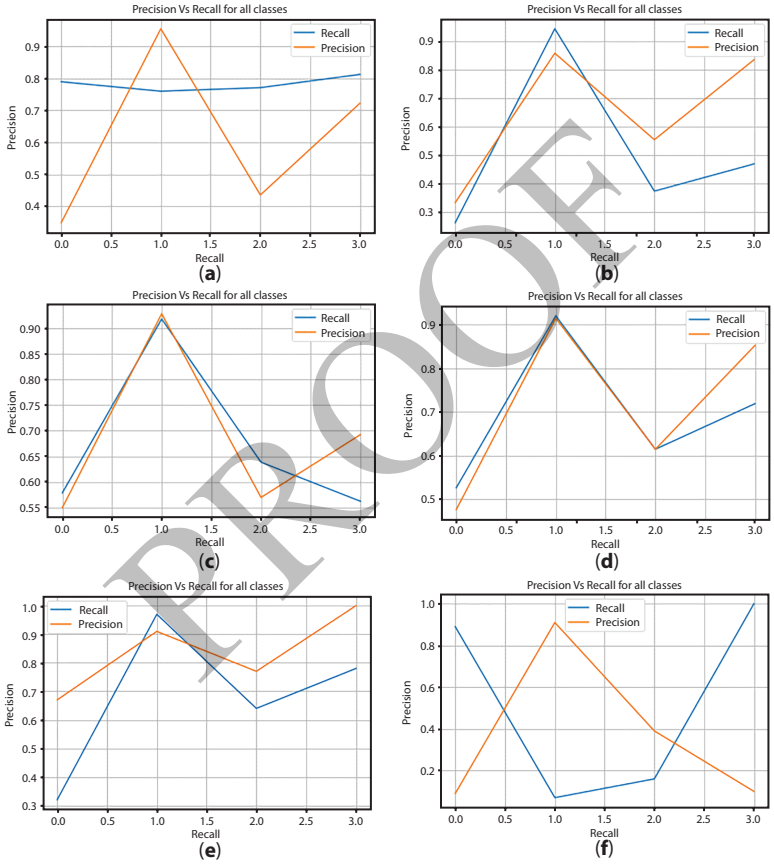

Figure 2.23 Precision and recall scores of machine learning methods for classification task. (a) Logistic regression. (b) K-nearest neighbours. (c) Decision tree. (d) Support vector classification. (e) Random forest. (f) Naïve bayes.

From Figure 2.23, when compared to other classes, accuracy and recall are shown to be greater for Class 1. This might be due to the fact that class 1 has more samples than the other classes. The other performance parameters are listed out in Table 2.4.

Table 2.4 Performance parameters of machine learning models for classification.

	Accuracy	Cohen kappa Score	Matthews Corrcoef	Average Precision	Average Recall	F1-Score
Logistic regression	76.54	51.74	54.78	61.54	78.32	68.92
K-Nearest Neighbours	81.33	43.56	44.80	64.42	51.19	57.04
Decision Tree	84.76	60.58	60.62	68.50	67.44	67.97
Support Vector Classification	85.27	61.08	61.09	71.41	69.48	70.43
Random Forest	89.21	68.79	69.61	83.75	67.75	74.90
Naïve Bayes	74.82	50.68	52.64	59.44	78.12	67.81

From the performance measures listed in Table 2.4, When comparing the performance parameters of the mentioned machine learning models, it is obvious that the random forest method outperforms the others. Three deep learning networks were employed in this study: feed forward, LSTM, and CNN-based models. The sparse categorical cross entropy loss, SGD optimizer, and accuracy metrics are utilised in all three models.

All the three models are trained for 200epocs and the performance metrics of the three models are listed in Table 2.5 and it is observed that Feed Forward Neural Network is performing well while compared with other classification-based models.

2.6 Suggestions

The article will be helpful for the readers to understand about machine and deep learning algorithms, for the convenience of readers the same dataset is used for both the regression and classification task. However, we suggest the readers to use a dataset, which is prominent for classification to perform the classification task and regression for performing regression tasks. Also, in our dataset, there was no missing data and imbalance in the classes was not identified and hence the results might be difference if we have a dataset with missing values or imbalanced classes.

2.7 Conclusion

Using a home price dataset, we attempted to evaluate the performance of machine learning and deep learning models. The primary objective of this work includes Evaluating the performances of the models using various performance metrics and hence the models were not optimized much to get better results which might lead our system to overfit or underfit and mislead the comparison studies. While evaluating the performance measures, such as MAE, MSE, RMSE, and R2 score, it was identified that Random forest algorithm performs better among Machine learning models and Tensorflow-Keras–based model performs better in Deep Learning Models. On the classification-based models, six performance parameters including Accuracy, Cohen Kappa Score, Matthews Correlation Coefficient, average precision and average recall were calculated and identified random forest classification and feed forward–based neural network was performing good among machine learning (ML) and deep learning (DL) models. It

Table 2.5 Performance parameters of deep learning models for classification.

	Accuracy	Cohen kappa Score	Matthews Corrcoef	Average Precision	Average Recall	F1-Score
Feed Forward Neural Network	89.52	69.54	70.68	83.89	68.64	75.23
LSTM	82.02	44.64	46.20	68.75	48.50	56.87
CNN	84.76	60.58	60.62	68.50	67.44	67.97

was also discovered that deep learning models outperformed machine learning models in both regression and classification tests.

References

1. Sewak, M., Sahay, S.K., Rathore, H., Comparison of Deep Learning and the Classical Machine Learning Algorithm for the Malware Detection, in: *19th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD)*, pp. 293–296, 2018.
2. Doleck, T., Lemay, D.J., Basnet, R.B., Bazelais, P., Predictive analytics in education: a comparison of deep learning frameworks. *Educ. Inf. Technol.*, 25, 3, 1951–1963, 2020. doi: 10.1007/s10639-019-10068-4.
3. Dong, B. and Wang, X., Comparison Deep Learning Method to Traditional Methods Using for Network Intrusion Detection, in: *8th IEEE International Conference on Communication Softw are and Networks Comparison*, pp. 581–585, 2016.
4. Liu, Y. *et al.*, Performance comparison of deep learning techniques for recognizing birds in aerial images, in: *Proceedings - 2018 IEEE 3rd International Conference on Data Science in Cyberspace, DSC 2018*, pp. 317–324, 2018, doi: 10.1109/DSC.2018.00052.
5. Delany, S.J., Chen, H., McKeever, S., A comparison of classical versus deep learning techniques for abusive content detection on social media sites, in: *Social Informatics*, pp. 117–133, 2018.
6. Turkoglu, I. and Alakus, T.B., Comparison of deep learning approaches to predict COVID-19 infection. *Chaos Solitons Fractals*, 140, 1–8, November 2020.
7. Ghosalkar, N.N. and Dhage, S.N., Real Estate Value Prediction Using Linear Regression, in: *Proceedings - 2018 4th International Conference on Computing, Communication Control and Automation, ICCUBEA 2018*, pp. 1–5, 2018, doi: 10.1109/ICCUBEA.2018.8697639.
8. Phan, T.D., Housing price prediction using machine learning algorithms: The case of Melbourne city, Australia, in: *Proceedings - International Conference on Machine Learning and Data Engineering, iCMLDE 2018*, pp. 8–13, 2019, doi: 10.1109/iCMLDE.2018.00017.
9. Nahib, I., Suryanta, J., Analysis, R., Daoud, J., II, Real estate value prediction using multivariate regression models Real estate value prediction using multivariate regression models. *IOP Conf. Ser. Mater. Sci. Eng.*, 4, 1–7, 2017. doi: 10.1088/1757-899X/263/4/042098.
10. Varma, A., House Price Prediction Using Machine Learning And Neural Networks, in: *Second International Conference on Inventive Communication and Computational Technologies (ICICCT)*, pp. 1936–1939, 2020, doi: 10.1109/ICICCT.2018.8473231.

11. Madhuri, C.H.R., Anuradha, G., Pujitha, M.V., House Price Prediction Using Regression Techniques : A Comparative Study, in: *IEEE 6th International Conference on smart structures and systems ICSSS 2019. House*, pp. 1–5, 2019, doi: 10.1109/ICSSS.2019.8882834.
Q3 12. Kashyap, I., Panda, S.P., Bansal, U., Narang, A., Sachdeva, A., Empirical analysis of regression techniques by house price and salary prediction Empirical analysis of regression techniques by house price and salary prediction. *IOP Conf. Ser. Mater. Sci. Eng.*, 1022, 2021. doi: 10.1088/1757-899X/1022/1/012110.
13. Rawool, A.G., Rogye, D.V., Rane, S.G., Vinayk, A., House Price Prediction Using Machine Learning. *IRE Journals*, 4, 11, 29–33, 2021.
14. Kaggle, House Prices - Advanced Regression Techniques. https://www.kaggle.com/c/house-prices-advanced-regression-techniques/data (accessed Jun. 10, 2021).
15. Bold BI, Data Visualization: Importance and Benefits. https://www.boldbi.com/blog/data-visualization-importance-and-benefits (accessed Jun. 10, 2021).
16. Analytiks, Why Data Visualization Is Important. https://analytiks.co/importance-of-data-visualization/ (accessed Jun. 10, 2021).
17. Histogram Definition. https://www.investopedia.com/terms/h/histogram.asp (accessed Jun. 10, 2021).
18. Statistics How To, Box Plot (Box and Whiskers): How to Read One & How to Make One in Excel, TI-83, SPSS. https://www.statisticshowto.com/probability-and-statistics/descriptive-statistics/box-plot/ (accessed Jun. 10, 2021).
Q4 19. Quantile-Quantile Plot, https://www.itl.nist.gov/div898/handbook/eda/section3/qqplot.htm (accessed Jun. 10, 2021).
20. Scatter Plot - Overview. https://corporatefinanceinstitute.com/resources/knowledge/other/scatter-plot/ (accessed Jun. 10, 2021).
21. Countplot. https://seaborn.pydata.org/generated/seaborn.countplot.html (accessed Jun. 10, 2021).
22. Data Preparation for Machine Learning. https://www.datarobot.com/wiki/data-preparation/ (accessed Jun. 10, 2021).
23. Al-Helali, B., Chen, Q., Xue, B., Zhang, M., A Hybrid GP-KNN Imputation for Symbolic Regression with Missing Values, in: *AI 2018: Advances in Artificial Intelligence*, pp. 345–357, 2018.
Q5 24. Santurkar, S., Tsipras, D., Ilyas, A., Madry, A., How does batch normalization help optimization? *Adv. Neural Inf. Process. Syst.*, 2483–2493, 2018. 1805.11604.
25. Potdar, K., Pardawala, T.S., Pai, C.D., A Comparative Study of Categorical Variable Encoding Techniques for Neural Network Classifiers. *Int. J. Comput. Appl.*, 175, 4, 7–9, 2017. doi: 10.5120/ijca2017915495.
Q6 26. Vining, G., Montgomery, D.C., Peck, E.A., *Introduction to Linear Regression Analysis*, John Wiley & Sons, 2012.
27. Caramiaux, B. and Tanaka, A., Machine Learning of Musical Gestures. *Proc. Int. Conf. New Interfaces Music. Expr. 2013 (NIME 2013*, pp. 513–518, 2013, [Online]. Available: http://nime2013.kaist.ac.kr/.

Q3
28. Li, L., Chen, S., Yang, C., Meng, F., Sigrimis, N., Prediction of plant transpiration from environmental parameters and relative leaf area index using the random forest regression algorithm. *J. Clean. Prod.*, 261, 2020. doi: https://doi.org/10.1016/j.jclepro.2020.121136.
29. Sarkar, A., Sai, K.K., Prakash, A., Veera, G., Sai, V., Kaur, M., A Research Paper on Loan Delinquency Prediction. *Int. Res. J. Eng. Technol.*, 8, 4, 715–722, 2021.
30. Taherkhani, A., Cosma, G., McGinnity, T.M., AdaBoost-CNN: An adaptive boosting algorithm for convolutional neural networks to classify multi-class imbalanced datasets using transfer learning. *Neurocomputing*, 404, 351–366, 2020. doi: 10.1016/j.neucom.2020.03.064.
31. Boosting Algorithms Explained. https://towardsdatascience.com/boosting-algorithms-explained-d38f56ef3f30 (accessed Jun. 10, 2021).
32. Cai, J., Xu, K., Zhu, Y., Hu, F., Li, L., Prediction and analysis of net ecosystem carbon exchange based on gradient boosting regression and random forest. *Appl. Energy*, 262, 114566, 2020, doi: https://doi.org/10.1016/j.apenergy.2020.114566.
33. Gradient Boosting Regression. http://deepnote.me/2019/08/25/datascience-18-machine-learning-with-tree-based-models-in-python/ (accessed Jun. 11, 2021).
34. Jerrita, S., Sajeev Ram, S., Haribaabu, V., Arun, S., ANALYSIS OF FILTERS IN ECG SIGNAL FOR EMOTION PREDICTION. *J. Adv. Res. Dyn. Control Syst.*, 12, 04, 896–902, 2020. doi: 10.5373/JARDCS/V12SP4/20201559.
35. medium.com, Support Vector Regression. https://medium.com/essence-of-learning/intuition-behind-support-vector-regression-3601f670a2ef (accessed Jun. 11, 2021).
36. Pujari, S., Ramakrishna, A., Padal, K.T.B., Comparison of ANN and Regression Analysis for Predicting the Water Absorption Behaviour of Jute and Banana Fiber ReinforcedEpoxy composites. *Mater. Today Proc.*, 4, 2, Part A, 1626–1633, 2017. doi: https://doi.org/10.1016/j.matpr.2017.02.001.
37. Artificial Neural Network. https://www.javatpoint.com/artificial-neural-network (accessed Jun. 11, 2021).
38. Bilgili, M., Tosun, E., Aydin, K., Comparison of linear regression and artificial neural network model of a diesel engine fueled with biodiesel-alcohol mixtures. *Alex. Eng. J.*, 5, 4, 3081–3089, 2016. doi: https://doi.org/10.1016/j.aej.2016.08.011.
39. Xu, D., Shi, Y., Tsang, I.W., Ong, Y.S., Gong, C., Shen, X., Survey on Multi-Output Learning. *IEEE Trans. Neural Netw. Learn. Syst.*, 31, 7, 2409–2429, 2020. doi: 10.1109/TNNLS.2019.2945133.
40. Multi-output Regression. https://towardsdatascience.com/chained-multi-output-regression-solution-with-scikit-learn-4f44bf9c8c5b (accessed Jun. 12, 2021).

Q6
41. Géron, A., *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*, O'Reilly Media, 2019.

42. Basis, R., Classifier, F., Mapping, S., A Comparative Study of Kernel Logistic Regression, Radial Basis Function Classifier, Multinomial Naïve Bayes, and Logistic Model Tree for Flash Flood Susceptibility Mapping. *Water*, 12, 1, 239–260, 2020.
43. Logistic Regression. https://towardsdatascience.com/binary-classification-with-logistic-regression-31b5a25693c4 (accessed Jun. 12, 2021).
44. Rau, C.S. *et al.*, Prediction of mortality in patients with isolated traumatic subarachnoid hemorrhage using a decision tree classifier: A retrospective analysis based on a trauma registry system. *Int. J. Environ. Res. Public Health*, 14, 11, 1–10, 2017. doi: 10.3390/ijerph14111420.
45. Decision Tree Algorithm. https://www.kdnuggets.com/2020/01/decision-tree-algorithm-explained.html (accessed Jun. 12, 2021).
46. Mursalin, M., Zhang, Y., Chen, Y., Chawla, N.V., Automated epileptic seizure detection using improved correlation-based feature selection with random forest classifier. *Neurocomputing*, 241, 204–214, 2017. doi: https://doi.org/10.1016/j.neucom.2017.02.053.
47. Random Forest Classification. https://medium.com/swlh/random-forest-classification-and-its-implementation-d5d840dbead0 (accessed Jun. 12, 2021).
48. Feng, X., Li, S., Yuan, C., Zeng, P., Sun, Y., Prediction of Slope Stability using Naive Bayes Classifier. *KSCE J. Civ. Eng.*, 22, 3, 941–950, 2018. doi: 10.1007/s12205-018-1337-3.
49. Singh, A., Halgamuge, M.N., Lakshmiganthan, R., Impact of Different Data Types on Classifier Performance of Random Forest, Naïve Bayes, and K-Nearest Neighbors Algorithms. *Int. J. Adv. Comput. Sci. Appl.*, 8, 12, 1–11, 2017. doi: 10.14569/ijacsa.2017.081201.
50. Edureka, KNN Algorithm. https://www.edureka.co/blog/k-nearest-neighbors-algorithm/ (accessed Jun. 13, 2021).
51. Brilliant.org., Feedforward Neural Networks. https://brilliant.org/wiki/feed-forward-neural-networks/ (accessed Jun. 13, 2021).
52. Chen, X.-Y. and Chau, K.-W., Uncertainty Analysis on Hybrid Double Feedforward Neural Network Model for Sediment Load Estimation with LUBE Method. *Water Resour. Manag.*, 33, 10, 3563–3577, 2019. doi: 10.1007/s11269-019-02318-4.
53. Raj, J.S. and Ananthi, J.V., Recurrent Neural Networks and LSTM explained. *J. Soft Comput. Paradig.*, 01, 01, 33–40, 2019.
54. Boufeloussen, O. and Medium, Recurrent Neural Network (RNN). https://medium.com/swlh/simple-explanation-of-recurrent-neural-network-rnn-1285749cc363 (accessed Jun. 13, 2021).
55. Pydeeplearning, Architecture of LSTM. https://pydeeplearning.weebly.com/blog/basic-architecture-of-rnn-and-lstm (accessed Jun. 13, 2021).

Answer all Queries (Q) in the margin of the text. When requested, provide missing intext reference citation as well as intext reference for figure/table. Please annotate the PDF and read the whole chapter again carefully. Careful proofing is key to optimal output.

Author Queries

Q1 Author names were inserted in the references actively cited. Please confirm that inserted names are correct.

Q2 Please provide journal title, year, volume and page number of references 14, 15, 16, 17, 18, 20, 21, 22, 31, 33, 35, 37, 40, 43, 45, 47, 50, 51, 54, 55.

Q3 Please provide page number of references 12, 28.

Q4 Please provide year, publisher name and location of reference 19.

Q5 Please provide volume number of reference 24.

Q6 Please provide publisher location of references 26, 41.