Analysis of different Classifier algorithms in Machine Learning

Fardin Bin Rahman
20101592
Department of Computer Science and Engineering
fardin.bin.rahman@g.bracu.ac.bd

Shahriar Ahmed
20101588
Department of Computer Science and Engineering
shahriar.ahmed@g.bracu.ac.bd

Razin Rayan Rahat
20101001
Department of Computer Science and Engineering
razin.rayan.rahat@g.bracu.ac.bd

Ilmy Islam
20201214
Department of Computer Science and Engineering
ilmy.islam@g.bracu.ac.bd

***Abstract*—In recent years, e-commerce has exploded in popularity due to the speed and convenience of exchanging goods and global services. It makes an attempt to explain the principle behind e-commerce, business models designed specifically for e-commerce, and the merits of each as well as its inherent restrictions and limits. It comes to the conclusion that engaging in e-commerce affords a number of benefits to the various stakeholders. Researchers have conducted several experiments to make E-commerce sites more feasible and accessible to the customers using machine learning. Similarly, we have also experimented and implemented several supervised learning algorithms to make price predictions, among them Decision Tree Classifiers and Random Forest Classifiers both gave 95.05% accuracy score.**

***Index Terms*—E-commerce, kNN, Naïve Bayes, Decision Tree, Random Forest Classifier, SVC, Linear Regression, Ridge Regression**

Introduction

A subfield of artificial intelligence (AI) and computer science called machine learning focuses on the use of data and algorithms to model how humans learn, gradually increasing the accuracy of the system. The rapidly growing field of data science includes machine learning as a key element. Algorithms are trained using statistical techniques to produce classifications or predictions and to find important insights in data mining projects. The decisions made as a result of these insights influence key growth indicators in applications and enterprises, ideally. In addition, certain machine learning [1] algorithms have extremely specific applications; yet, the three primary approaches are still in use today. Only supervised learning techniques and classifications were used in this research. Additionally, we have decided to complete this project with an E-commerce dataset. In our research, we mainly concentrated on predicting the Selling Price of different products on an E-commerce site based on their Brand name, MRP, and Discount; by calculating the Selling Price with various algorithms and the accuracy score of those corresponding algorithms, we evaluated which classifier performs best for this type of dataset. Then we processed, cleaned up, and deleted redundant data. Once the preprocessing phase was finished, we utilized 7 different supervised machine learning classifiers to predict and achieve the highest level of accuracy. Our goal is to compare different types of supervised learning algorithm's efficiency and how they performed on a large dataset. In order to achieve our goal in this research, Scikit-Learn, often known as sklearn, is a free machine learning package for the Python programming language that we used to implement the supervised machine learning classifiers. For data preprocessing and visualization, additional free Python tools including Pandas, NumPy, and Matplotlib were used.

Literature Review

kNN

KNN is a method of case-based learning that retains all training data for classification. As a way of unmotivated learning, it is inapplicable to many applications, such as dynamic web mining for a big repository. Finding some representatives to represent the entire training data for classification is one technique to increase its efficacy. constructing an inductive learning model from the training dataset and classifying using this model (representatives). There are a number of current algorithms, such as decision trees and neural networks, that were originally intended to construct such a model. One of the criteria for evaluating algorithms is their performance. As kNN is a basic but effective classification method, and it is persuasive as one of the most effective methods in our situation, we consider it to be one of the most effective methods. [2]

Naïve Bayes

Naïve Bayes (NB) is a well-known probabilistic technique for classifying data. It is a simple yet effective algorithm with numerous real-world applications, including product suggestions, medical diagnostics, and autonomous vehicle control. Due to the inability of real data to satisfy the assumptions of NB, modifications of NB exist to accommodate general data. The Naïve Bayes approach has proven to be a practical and effective classification technique for multivariate data. However, characteristics are typically correlated, which violates the concept of conditional independence of the Naïve Bayes technique and may degrade its effectiveness. In addition, datasets frequently have a high number of characteristics, which can confuse the interpretation of the results and slow down the execution of the procedure. [3]

Decision Tree

Decision trees are a strong technique utilized in numerous domains, including machine learning, image processing, and pattern recognition. DT is a model that efficiently and cohesively combines a number of fundamental tests in which a numerical characteristic is compared to a threshold value in each test. The conceptual principles in a neural network of connections between nodes are significantly simpler to create than the numerical weights. DT is utilized for grouping purposes primarily. In addition, DT is a prevalent classification model in Data Mining. Each tree is comprised of its nodes and branches. Each node represents features in a category to be classified, whereas each subset defines a possible value for each node. [4]

Random Forest Classifier

Random Forest is one of the flexible, straightforward, and hyperparameter-free supervised learning methods. This classification system is highly good. In Random Forest, there is a minimum number of trees that must be constructed in order to classify all available data. Where the amount depends heavily on each data. The minimal number of trees for each data set is affected by the amount of breaker properties. The number of trees has a significant effect on the level of precision. Beginning with the smallest number of trees, increasing the number of trees produces greater accuracy. There is an optimal level of accuracy, after which accuracy is attained even while the number of trees and accuracy stay unchanged. The number of breaker attributes impacts this algorithm's precision. The Random Forest with a number of breaker attributes equal to the number of accessible attributes will have low precision. [5]

Support Vector Classifier

Support Vector Machine, or SVM, is one of the most used techniques for Classification and Regression issues in Supervised Learning. In Machine Learning, it is used mostly for Classification issues. The objective of the SVM algorithm is to generate the optimal line or decision boundary that divides n-dimensional space into classes, so that subsequent data points can be easily classified. This optimal decision boundary is referred to as a hyperplane. SVM selects the extreme points/vectors that contribute to the formation of the hyperplane. These extreme situations are known as support vectors, and the corresponding method is known as the Support Vector Machine.

Linear regression

The linear regression algorithm is one of the simplest and most prevalent Machine Learning algorithms. This statistical technique is utilized for predictive analysis. Linear regression predicts continuous/real or numerical variables such as sales, salary, age, and product price, among others.

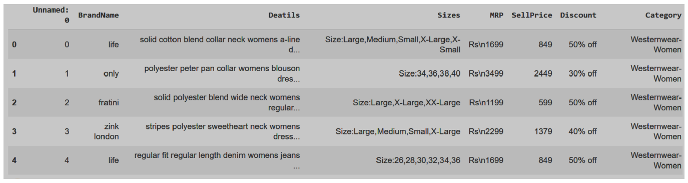
The algorithm for linear regression demonstrates a linear relationship between a dependent variable (y) and one or more independent variables (y), hence the name linear regression. As linear regression demonstrates a linear relationship, it determines how the value of the dependent variable varies in proportion to the value of the independent variable.

Ridge Regression

Ridge regression is a statistical technique for predicting the coefficients of multiple-regression models in instances where the independent variables are strongly correlated, in comparison to linear regression, which is the industry standard algorithm for regression and assumes a linear relationship between input variables and the target variable. Ridge Regression is a linear regression improvement that, during training, effectively nullifies the loss function for regularization.

DATASET

Dataset Description



	Unnamed: 0	BrandName	Deatils	Sizes	MRP	SellPrice	Discount	Category
0	0	life	solid cotton blend collar neck womens a-line d...	Size:Large,Medium,Small,X-Large,X-Small	Rs\n1699	849	50% off	Westernwear-Women
1	1	only	polyester peter pan collar womens blouson dres...	Size:34,36,38,40	Rs\n3499	2449	30% off	Westernwear-Women
2	2	fratini	solid polyester blend wide neck womens regular...	Size:Large,X-Large,XX-Large	Rs\n1199	599	50% off	Westernwear-Women
3	3	zink london	stripes polyester sweetheart neck womens dress...	Size:Large,Medium,Small,X-Large	Rs\n2299	1379	40% off	Westernwear-Women
4	4	life	regular fit regular length denim womens jeans ...	Size:26,28,30,32,34,36	Rs\n1699	849	50% off	Westernwear-Women

Fig. 1. Reading the Dataset

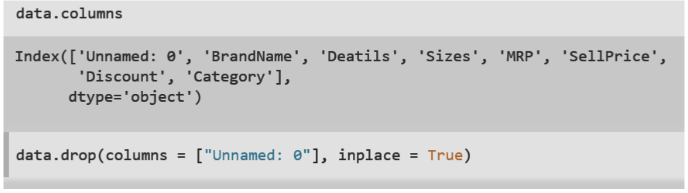
We first generated a Pandas Dataframe after reading the dataset from the csv file. There are a total of 8 columns displayed here, and the column ”Unnamed:0” is one of them. This column is unnecessary, and we want to get rid of it during the data preprocessing stage. Regarding the dataset:

BrandName: Mentions the brand of the product
Details: Details about the product
Size: Sizes available
MRP: This is max retail price
SellPrice: This is the price after discount

Category: Category of the product
Nan value is null value

Data Processing

At first we will delete the unnecessary column named “Unnamed: 0” from our dataframe.
Then we further drop all the rows that contain Nan values



```
data.columns
```

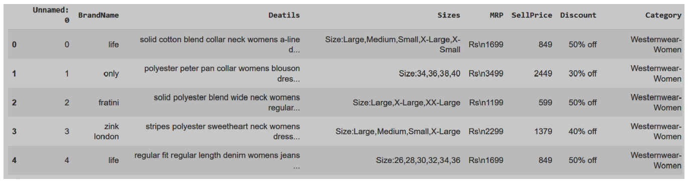
```
Index(['Unnamed: 0', 'BrandName', 'Deatils', 'Sizes', 'MRP', 'SellPrice',
       'Discount', 'Category'],
      dtype='object')
```

```
data.drop(columns = ["Unnamed: 0"], inplace = True)
```

Fig. 2. Deleting the Unnamed: 0 column

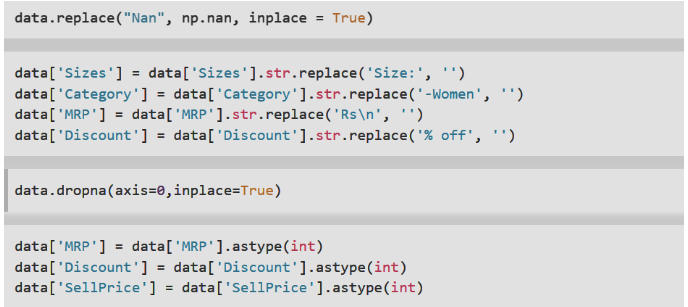
from our dataframe; moreover, we formatted the strings of the columns so that we can eradicate the redundant data as those data can cause various errors and problems in our research.

:



	Unnamed: 0	BrandName	Deatils	Sizes	MRP	SellPrice	Discount	Category
0	0	life	solid cotton blend collar neck womens a-line d...	Size:Large,Medium,Small,X-Large,X-Small	Rs\n1699	849	50% off	Westernwear-Women
1	1	only	polyester peter pan collar womens blouson dres...	Size:34,36,38,40	Rs\n3499	2449	30% off	Westernwear-Women
2	2	fratini	solid polyester blend wide neck womens regular...	Size:Large,X-Large,XX-Large	Rs\n1199	599	50% off	Westernwear-Women
3	3	zink london	stripes polyester sweetheart neck womens dress...	Size:Large,Medium,Small,X-Large	Rs\n2299	1379	40% off	Westernwear-Women
4	4	life	regular fit regular length denim womens jeans ...	Size:26,28,30,32,34,36	Rs\n1699	849	50% off	Westernwear-Women

Fig. 3. Before formatting Strings.



```
data.replace("Nan", np.nan, inplace = True)
```

```
data['Sizes'] = data['Sizes'].str.replace('Size:', '')
data['Category'] = data['Category'].str.replace('-Women', '')
data['MRP'] = data['MRP'].str.replace('Rs\n', '')
data['Discount'] = data['Discount'].str.replace('% off', '')
```

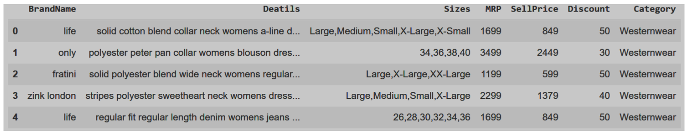
```
data.dropna(axis=0,inplace=True)
```

```
data['MRP'] = data['MRP'].astype(int)
data['Discount'] = data['Discount'].astype(int)
data['SellPrice'] = data['SellPrice'].astype(int)
```

Fig. 4. Formatting the strings.

:

After removing redundancy from our dataframe we plot a correlation graph and scatter matrix between a few columns.



	BrandName	Deatils	Sizes	MRP	SellPrice	Discount	Category
0	life	solid cotton blend collar neck womens a-line d...	Large,Medium,Small,X-Large,X-Small	1699	849	50	Westernwear
1	only	polyester peter pan collar womens blouson dres...	34,36,38,40	3499	2449	30	Westernwear
2	fratini	solid polyester blend wide neck womens regular...	Large,X-Large,XX-Large	1199	599	50	Westernwear
3	zink london	stripes polyester sweetheart neck womens dress...	Large,Medium,Small,X-Large	2299	1379	40	Westernwear
4	life	regular fit regular length denim womens jeans ...	26,28,30,32,34,36	1699	849	50	Westernwear

Fig. 5. Dataframe after formatting strings and columns.

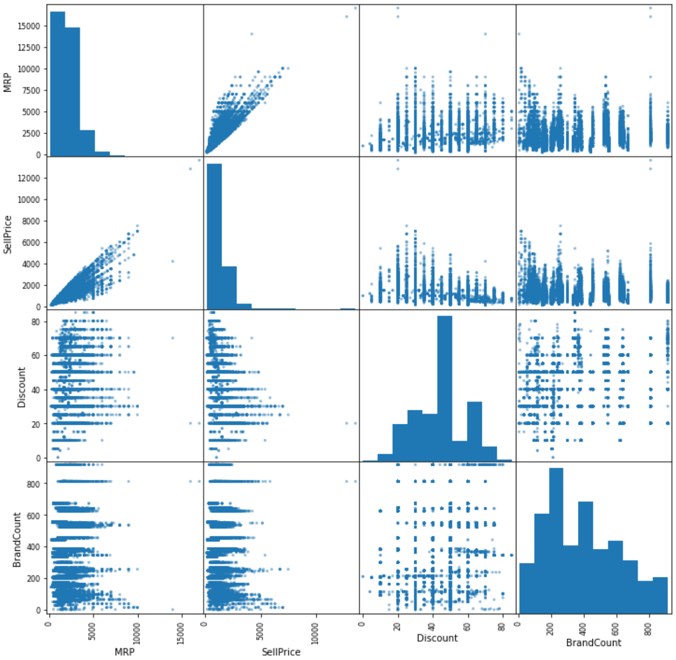


Fig. 6. Scatter Matrix

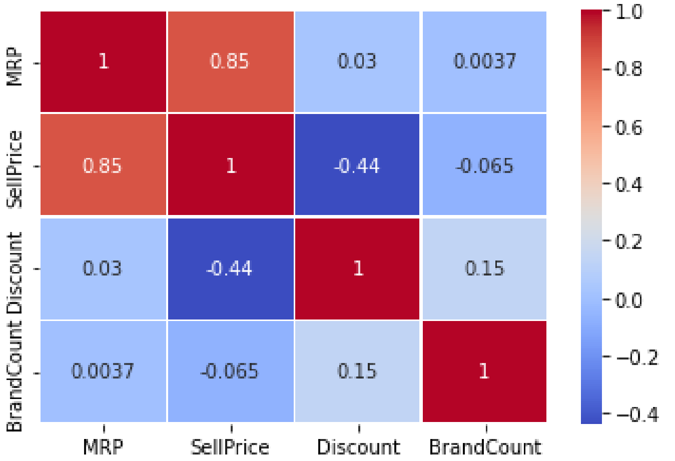


Fig. 7. Correlation Graph

Then we plotted a scatter plot of MRP and Selling Price based on the Category column.

We convert BrandName, MRP, Discount and Selling Price as String type data and we tried to predict the selling price of each product based on their BrandName, MRP and Discount. Then, we divide the dataset, we take BrandName, MRP and Discount as X and Selling Price as y. Using train_test_split from sklearn we split the dataset into two parts, train dataset and test dataset. 20% of the dataset was selected for the test dataset.

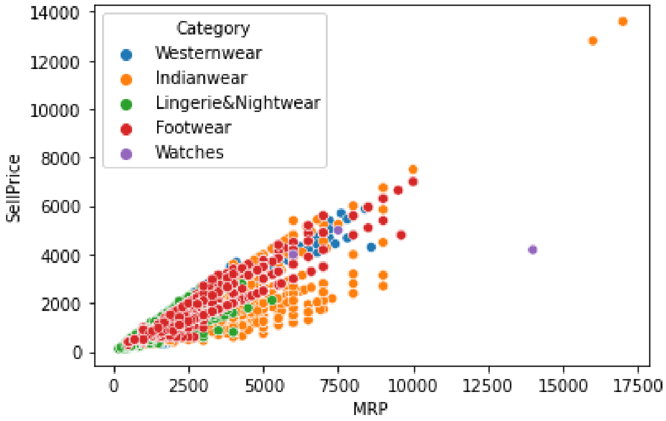
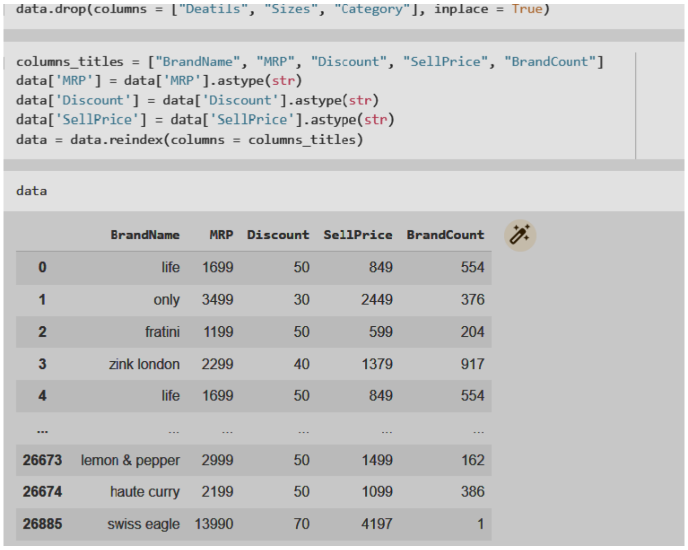


Fig. 8. Scatter Plot



```
data.drop(columns = ["Deatils", "Sizes", "Category"], inplace = True)
```

```
columns_titles = ["BrandName", "MRP", "Discount", "SellPrice", "BrandCount"]
data['MRP'] = data['MRP'].astype(str)
data['Discount'] = data['Discount'].astype(str)
data['SellPrice'] = data['SellPrice'].astype(str)
data = data.reindex(columns = columns_titles)
```

```
data
```

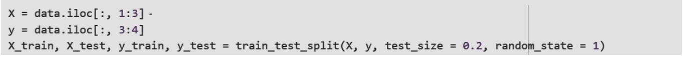
	BrandName	MRP	Discount	SellPrice	BrandCount
0	life	1699	50	849	554
1	only	3499	30	2449	376
2	fratini	1199	50	599	204
3	zink london	2299	40	1379	917
4	life	1699	50	849	554
...	...	...	...	...	...
26673	lemon & pepper	2999	50	1499	162
26674	haute curry	2199	50	1099	386
26885	swiss eagle	13990	70	4197	1

Fig. 9. Processing the Columns

METHODOLOGY

kNN

The KNN algorithm believes that related things are located nearby. In other terms, ”Birds of a feather flock together” or comparable objects are close to one another. KNN uses the arithmetic we may have learned as children—calculating the distance between points on a graph—to encapsulate the idea of similarity (also called distance, proximity, or closeness). First we fill the data up. Set K to the number of neighbors chosen. For each example in the data, we determine from the data the distance between the query example and the current example. To an ordered collection, add the example's distance and index. Arrange the distances in ascending order from smallest to largest in the ordered collection of indices and distances. Choose the first K items from the collection that has been sorted. Obtain the labels of the K entries that you chose. If classification, give the K labels' mode. [6]



```
X = data.iloc[:, 1:3]
y = data.iloc[:, 3:4]
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2, random_state = 1)
```

Fig. 10. Train Test Splitting

Naïve Bayes

The Naive Bayes classifier works on the principle of conditional probability, as stated by the Bayes theorem. This theorem is divided into 5 parts. First we separate the dataset by class. Then we summarize the dataset then we summarize it by classes. Afterwards, Gaussian Probability Density Function is used. Then we calculate the Class Probabilities. We can frame classification as a conditional classification problem with Bayes Theorem as follows: [7]

$$P\left(yi|x1,x2,..,xn\right) = P\left(x1,x2,..,xn|yi\right) * P\left(yi\right)/P\left(x1,x2,..,xn\right)$$

Decision Tree

To decide whether to divide a node into two or more sub-nodes, decision trees employ a variety of techniques. The homogeneity of newly formed sub-nodes is increased by sub-node creation. In other words, we can claim that the node's purity improves in relation to the desired variable. With no backtracking, the top-down greedy search method of the ID3 algorithm creates decision trees by traversing the space of potential branches. As the name implies, a greedy algorithm always selects the option that, at the time, appears to be the best.

The root node is the original set S at the start. The approach determines the entropy (H) and information gain (IG) of the very unused attribute of the set S for each iteration. The property with the lowest entropy or highest information gain is then chosen. To create a subset of the data, the set S is then divided by the chosen attribute. Only traits that have never been chosen previously are taken into account as the algorithm iterates over each subset. [8]

Random Forest Classifier

Like its name suggests, a random forest is made up of numerous independent decision trees that work together as an ensemble. The class with the highest votes becomes the prediction made by our model. The random forest's individual trees each spit forth a class prediction. In this case, ensemble methods combine several learning algorithms to achieve higher predicted performance than any one of the individual learning algorithms could.

It selects random K data points from the training set and builds the decision trees associated with the selected data points (Subsets). Then it chooses the number N for decision trees that you want to build and Repeat Step 1 & 2. For new data points, find the predictions of each decision tree, and assign the new data points to the category that wins the majority votes. [9]

Support Vector Classifier

SVM categorizes data points even when they are not otherwise linearly separable by mapping the data to a high-dimensional feature space. Once a separator between the categories is identified, the data are converted to enable the hyperplane representation of the separator. The group to which a new record should belong can therefore be predicted using the features of new data.

VM can get away with just the dot products between them; it actually doesn't need the actual vectors to perform its magic. As a result, we may avoid performing the time-consuming calculations for the new dimensions. We employ a mechanism known as a ”Kernel Function” to do this. The Kernel's value is often set to ”Linear,” however it can also be different. [10]

Linear regression

Multiple Linear Regression is similar to simple linear regression but here we have more than one independent or explanatory variable. Linear Regression can be written mathematically as follows: [11]

$Y = \beta 0 + \beta 1X1 + \beta 2X2 + \beta 3X3 + \beta 4X4 + \beta 5X5 + \beta 6X6 + \epsilon$

$\beta 0 = Y - intercept(alwaysaconstant); \beta 1, \beta 2, \beta 3, \beta 4, \beta 5, \beta 6 =$ regression coefficients; $\epsilon = Errorterms(Residuals)$.

Finding the best fit line is crucial when using linear regression since it minimizes the difference between the predicted and actual values. The line with the least inaccuracy will have the best fit. We need to determine the best values for a0 and a1 to find the best fit line, thus to do this we utilize the cost function. The varied values for weights or the coefficient of lines (a0, a1) gives a distinct line of regression.

Ridge Regression

An optimization procedure is used to find the model's coefficients in order to reduce the total squared error between the predictions (yhat) and the anticipated target values (y). [12]

$loss = sumi = 0ton(yi - \hat{y}i)^2$

A common punishment is to evaluate a model's performance according to the sum of its squared coefficient values (beta). The penalty is referred to as an L2.

$l2penalty = sumj = 0$ to p βj^2

Although it avoids any coefficients from being eliminated from the model by allowing their value to become zero, an L2 penalty reduces the size of all coefficients.

RESULT ANALYSIS

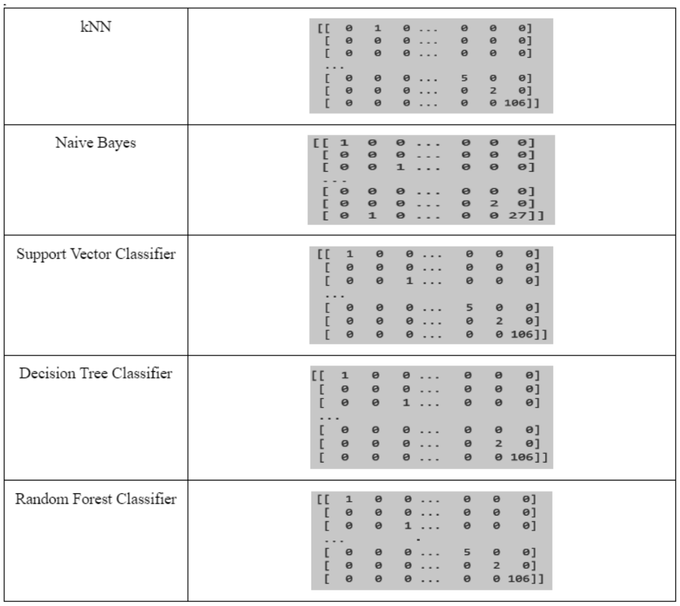
There have been in total 7 different supervised learning algorithms used in the research to analyze which model is more efficient, accurate and performs better incase of a large dataset. We have calculated Accuracy Score, Recall Score, Precision Score, F1 Score and Confusion Matrix(if needed) for every single classifier to check and analyze their performances on this large dataset.

Models	Accuracy Score	Recall	f1-score	Precision	Confusion Matrix
KNN	90.48%	90.48%	90.48%	90.48%	Y
Naive Bayes	37.77%	37.77%	37.77%	37.77%	Y
Decision Tree	95.05%	95.05%	95.05%	95.05%	Y
Random Forest Classifier	95.05%	95.05%	95.05%	95.05%	Y
Support Vector Classifier	94.61%	94.61%	94.61%	94.61%	Y
Linear Regression	93.69%	93.69%	93.69%	93.69%	N/A
Ridge Regression	93.69%	93.69%	93.69%	93.69%	N/A

From the table we can infer that the best accuracy scores have been provided by Decision Tree classifier and Random Forest Classifier followed by Support Vector Classifier, Linear Regression, Ridge Regression and kNN Classifier. While 6 of the 7 classifier gives an accuracy score of 90% or more than 90%; but Naive Bayes Classifier fails miserably in this case and gives roughly 38% accuracy score which further proves that unlike others, Naive Bayes Classifiers is not a good option when you are working with a really big dataset.

Confusion matrices were calculated for 5 out of 7 classifier algorithms and those are:

1) kNN
2) Naive Bayes
3) SVC
4) Decision Tree
5) Random Forest Classifier



kNN

```
[[  0   1   0 ...   0   0   0]
 [  0   0   0 ...   0   0   0]
 [  0   0   0 ...   0   0   0]
 ...
 [  0   0   0 ...   5   0   0]
 [  0   0   0 ...   0   2   0]
 [  0   0   0 ...   0   0 106]]
```

Naive Bayes

```
[[ 1  0  0 ...  0  0  0]
 [ 0  0  0 ...  0  0  0]
 [ 0  0  1 ...  0  0  0]
 ...
 [ 0  0  0 ...  0  0  0]
 [ 0  0  0 ...  0  2  0]
 [ 0  1  0 ...  0  0 27]]
```

Support Vector Classifier

```
[[  1   0   0 ...   0   0   0]
 [  0   0   0 ...   0   0   0]
 [  0   0   1 ...   0   0   0]
 ...
 [  0   0   0 ...   5   0   0]
 [  0   0   0 ...   0   2   0]
 [  0   0   0 ...   0   0 106]]
```

Decision Tree Classifier

```
[[  1   0   0 ...   0   0   0]
 [  0   0   0 ...   0   0   0]
 [  0   0   1 ...   0   0   0]
 ...
 [  0   0   0 ...   0   0   0]
 [  0   0   0 ...   0   2   0]
 [  0   0   0 ...   0   0 106]]
```

Random Forest Classifier

```
[[  1   0   0 ...   0   0   0]
 [  0   0   0 ...   0   0   0]
 [  0   0   1 ...   0   0   0]
 ...
 [  0   0   0 ...   5   0   0]
 [  0   0   0 ...   0   2   0]
 [  0   0   0 ...   0   0 106]]
```

Fig. 11. Confusion Matrix.

Some bars charts were also plotted using pyplot.bar method based on the accuracy, recall, precision and f1_score to analyze their performances.

Those charts are given below:

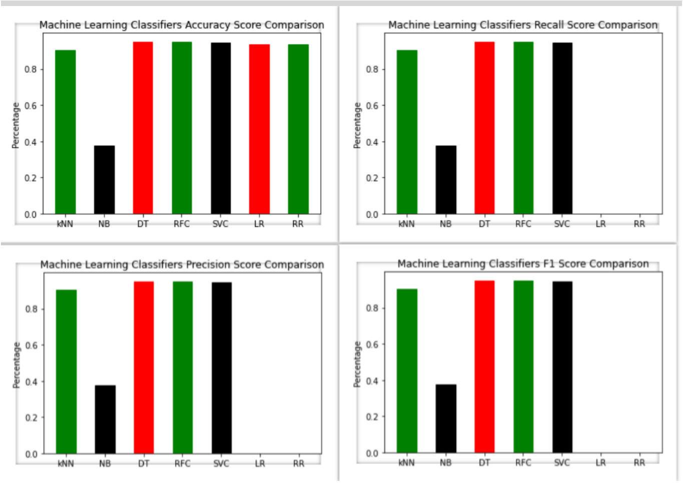


Fig. 12. Score Comparison.

References

[1] IBM Cloud Education (2020) ”Machine Learning”

[2] G. Guo, H. Wang, D. Bell, Y. Bi & K. Greer ”KNN Model-Based Approach in Classification”

[3] I. Wickramasinghe & H. Kalutarage ”Naive Bayes: applications, variations and vulnerabilities: a review of literature with code snippets for implementation”

[4] A. J. Myles, R. N. Feudale,Y. Liu, N. A. Woody, S. D. Brown ”An introduction to decision tree modeling”

[5] M. Huljanah, Z. Rustam, S. Utama and T. Siswantining ”Feature Selection using Random Forest Classifier for Predicting Prostate Cancer”

[6] O. Harrison ”Machine Learning Basics with the K-Nearest Neighbors Algorithm”

[7] Simplilearn ”Understanding Naive Bayes Classifier”

[8] N. S. Chauhan, ”Decision Tree Algorithm, Explained”

[9] T. Yiu ”Understanding Random Forest”

[10] IBM Documentation ”How SVM Works”

[11] S. Glen. ”Linear Regression: Simple Steps, Video. Find Equation, Coefficient, Slope” From StatisticsHowTo.com

[12] J. Brownlee ”How to Develop Ridge Regression Models in Python”