A Model for Prediction of Consumer Conduct Using Machine Learning Algorithm

Mr. B. Narasimha Rao, Assoc Prof. CSE, BVCE, bnraolak@gmail.com
Mr. Y. M. S. D. Sastry, Assoc Prof. CSE:BVCE, ymsdsastry.bvce@bvcgroup.in
Kamma Vanyasri, CSE:BVCE
Kanchustambham Venkayya Swamy, CSE:BVCE
Kavala Sarath Siva Chaitanya, CSE:BVCE
Gandrothu Nivas Ram, CSE:BVCE

Article Info
Page Number: 5213 - 5226
Publication Issue:
Vol 71 No. 4 (2022)

Article History
***Article Received**: 25 March 2022*
***Revised**: 30 April 2022*
***Accepted**: 15 June 2022*
***Publication**: 19 August 2022*

Abstract

Because machine learning algorithms are so good at making predictions, they have become very important. Due to unanticipated customer situations, it's hard to know how a customer will do. There are many algorithms that all do the same thing. In this paper, we looked at and analysed three Bays algorithms: AODE, Naive Bayes, and AODEsr. We put these algorithms into the WEKA tool and made a new model that is more accurate than the one we were using before. During development, we've tried to cut down on noise and mistakes in the data, and we also need to filter the information. This process will give the new data that has been filtered a weight of Wj. The error can be described as E(j,k), where j is an assumption and k is a function of purpose. In the same way, the function N = E + Wj can be used to describe noise.

INTRODUCTION

Artificial intelligence includes machine learning, in which we teach a machine to predict the value we want. During this training, we set up some rules or patterns, and our machine finds the pattern we set up. So, in Machine Learning, input information is made based on knowledge stored in a database. Since we want our system to be able to predict or pull useful information from a set of input data, we need to come up with an algorithm and pattern to get the information we need. After these two steps, which are the development of the algorithm and pattern, are done, the machine can do the following:

• Find, pull out, and summarise relevant information • Make predictions based on analytical data • Figure out how likely certain effects are to happen

• To adjust to specific changes on your own

1.1 Computers that learn:

Machine learning is a branch of artificial intelligence (AI) and computer science that tries to mimic how humans learn by using data and algorithms. Over time, it gets better and better at doing this.

It is an important part of the field of data science, which is getting bigger and bigger. Statistical methods are used to teach algorithms how to classify or make predictions. This helps data mining projects find key insights. These insights then help applications and businesses make decisions, which should affect key growth metrics. As big data continues to grow and expand, the market will need more data scientists to help figure out the most important business questions and then find the data to answer them.

1.2 TYPES OF MACHINE LEARNING:

Basically, these algorithms are used to find patterns and then come up with a solution. Algorithms for machine learning can be grouped as:

Learning with a Teacher:

In this kind of learning, all of the information is ready ahead of time. That should be explained so that groups of algorithms get the right amount of data. In other words, the system learns based on how much power it gets and how much it gives out. The programme manager, who acts as a kind of teacher, gives the right amount of feedback in supervised learning. The goal is to train the method from the point of view of sequentially calculating inputs and outputs and to set up communication. Based on the idea of autonomy, the Naive Bayes is a model for making probabilistic distinctions. In many real-world mining situations, however, this statement is often not true. In response to this claim, scholars have done a lot of work to test whether or not NB is right by making their stability worse.

Averaged One Dependence Estimators (AODE) is an idea that reduces the independent predictive value by taking samples from all prototypes in a limited class of dependent classifiers. Based on this research, we think that giving each of these different classifiers a different value can lead to better improvement. We have tested our algorithm with the Weka tool and the Super Market data sets. We have also briefly described a study that compares NaiveBayes, AODE, and AODEsr. The results of the research show that the proposed algorithm is significantly better than all the other algorithms that were used to compare Unsupervised Learning:

In this way of learning, values aren't given until later. It is mostly used to group things together. The machine tries to organise and sort the information you put in based on certain features. For example, a machine can learn that different-colored coins can be put in order by putting them in order by "colour."

1.3 Motivation:

The goal of this study is to find out how machine learning algorithms can help predict consumer behaviour better and how combining these models works better than existing ones, since there isn't a single paper that talks about the predictions made here. Lastly, it's important to know and understand how these models can be different from each other in predicting data.

1.4 Problem statement:

The main problem with this is that getting the data is hard and takes a long time.

1.5 The goal is to predict or get information from the input data, as well as to predict how consumers will act.

1.6 Work Scope:

The work's goals are

• Make predictions based on analytical data • Figure out how likely it is that certain things will happen

1.7 Possible Uses:

• Speech Recognition: This is the process of turning voice instructions into text. It is also called "Speech to text" or "Computer speech recognition."

• Traffic prediction: It predicts the traffic conditions, such as whether traffic is clear, moving slowly, or very backed up.

The Google Maps app and sensors give the real-time location of the car.

At the same time, average time has taken in the past.

• Product recommendations: Google uses different machine learning algorithms to figure out what the user is interested in and then suggests products based on what the customer is interested in.

3.1 Algorithm:

3.1.1 Algorithm for a random forest:

This model is based on three random ideas: picking training data at random when making trees, picking some subsets of features when splitting nodes, and only looking at a subset of all features when splitting each node in each simple decision tree. In a random forest, each tree learns from a random sample of the data points when it is being trained.

How to write an algorithm:
Step 1: begin
Step 2: Add the data values with their attributes.
Step 3: Check for noise and find out how often mistakes happen.
Step 4: Sort the data into groups based on how often they are wrong.
Step 5: Give the filtered and fixed data a weight, Wi. Step 6: Use the weighted data to test a model.
Step 7: Choose the model that fits the criteria.
Step 8: end

Bayes, naive:

When compared to more complex algorithms, the Naive Bayes classifier can be very fast. Because the class distributions aren't mixed up with each other, each one can be evaluated on its own as a one-dimensional distribution. In turn, this helps get rid of some of the problems caused by the "dimensionality curse."

The Nave Bayes classifier is a type of probabilistic classifier that is based on Bayes' theorem and assumes that the features are very independent from each other based on the value of the class variable. This method is a group of algorithms for learning with help.

The Decision Tree algorithm:

The goal is to create a model that can predict a target value by learning simple rules for making decisions based on the data features. This method has some benefits, like being easy to understand and explain or being able to solve problems with more than one output. A decision tree is a common way to use supervised learning for both regression and classification problems. The goal of a technique is to predict a goal by making simple rules for making decisions based on a dataset and related features. Two benefits of using this model are that it is easy to understand and can solve problems with different outcomes. On the other hand, one disadvantage is that it can lead to overfitting by building trees that are too complicated.

SVM: The Support Vector Machine (SVM) is a machine learning algorithm that learns from being watched. This model does a better job of predicting in the short and medium term than it does in the long term. Every algorithm has its own way of recognising patterns and making predictions based on them. The SVM model was used to look at the weekly trend of the NIKKEI 225 index to see how well it could predict financial trends. SVM is a boundary that uses a line or hyperplane to separate two classes in the best way. Equation shows how to find the decision boundary. SVMs use kernel functions like linear, non-linear, sigmoid, radial basis function (RBF), and polynomial to turn classes that can't be split into ones that can.

Logistic regression is a way to put observations into a set of classes that are different from each other. With the logistic sigmoid function, the output of the algorithm is changed into a probability value, which is then used to predict the target. Logistic Regression is like Linear Regression, but it uses a sigmoid function instead of a logistic function, which makes it more complicated. Logical regression is based on the idea that the cost function should only be between 0 and 1.

Bagging Classifier: A Bagging classifier is an ensemble meta-estimator that fits base classifiers to random subsets of the original dataset and then combines their individual predictions (either by voting or by averaging) to make a final prediction. Usually, this kind of meta-estimator is used to reduce the variance of a black-box estimator, like a decision tree, by adding randomness to its construction process and then making an ensemble out of it.

3.3 Architecture/Framework:

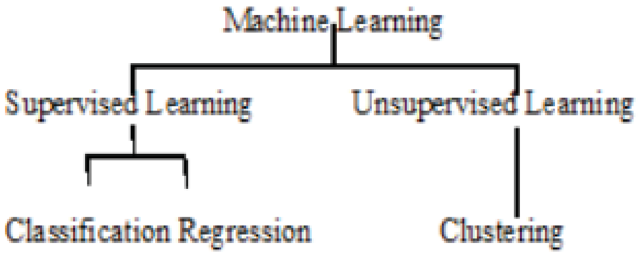


Fig.1 Framework

3.4Algorithm and Process Design:

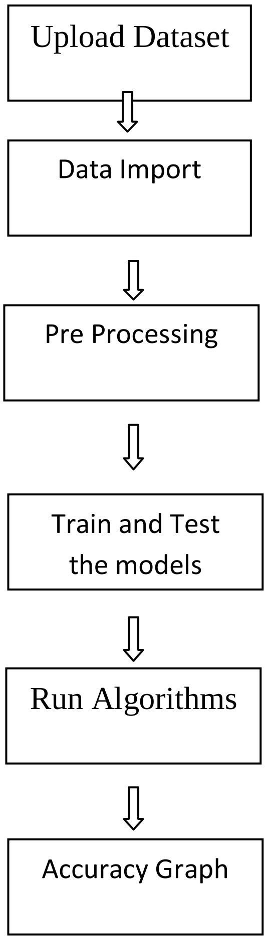
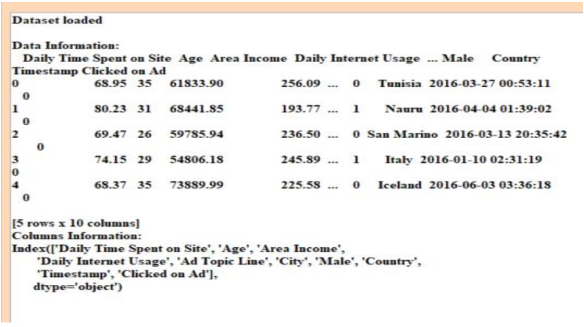


Fig.2. Process Design

4 Implementation:

4.1 About the Data:

In this we use consumer dataset data.csv in this we can get the data of daily internet usage, daily time spent on that site, city.this is downloaded for kaggle and uci repository



```
Dataset loaded

Data Information:
  Daily Time Spent on Site  Age  Area Income  Daily Internet Usage  ... Male    Country
Timestamp Clicked on Ad
0              68.95  35   61833.90           256.09 ...  0   Tunisia  2016-03-27 00:53:11
  0
1              80.23  31   68441.85           193.77 ...  1     Nauru  2016-04-04 01:39:02
  0
2              69.47  26   59785.94           236.50 ...  0 San Marino  2016-03-13 20:35:42
    0
3              74.15  29   54806.18           245.89 ...  1     Italy  2016-01-10 02:31:19
0
4              68.37  35   73889.99           225.58 ...  0   Iceland  2016-06-03 03:36:18
  0

[5 rows x 10 columns]
Columns Information:
Index(['Daily Time Spent on Site', 'Age', 'Area Income',
    'Daily Internet Usage', 'Ad Topic Line', 'City', 'Male', 'Country',
    'Timestamp', 'Clicked on Ad'],
   dtype='object')
```

4.2 Performance Metrics:

Confusion Matrix: A confusion matrix, which is also called an error matrix, is used in the field of machine learning to solve the problem of statistical classification. It is a square grid whose size is determined by how many classes your model has.

Precision is the number of correct positive predictions out of the total number of positive predictions. Accuracy = TP/TP + FP

Remember that it is the number of positive observations that can be predicted correctly out of all the observations in the original data. Recall = TP/TP plus FN

F1-score: It is the average of Precision and Recall with a weighted average. So, both false positives and false negatives are taken into account in this score. It's not as easy to understand as accuracy, but this is usually more useful than accuracy, especially if you have a rough idea of how the students in a class are spread out. The best way for accuracy to work is if both false positives and false negatives cost the same. If the prices for Precision and Recall are different, it's best to look at both.

F1 Score = 2(Precision Recall/Precision + Recall)

ROC Area: A receiver operating characteristic curve (ROC curve) is a graph that shows how well a classification model works at every classification threshold. This curve shows two different values:

- Rate of True Positives
- Rate of False Positives

Rate of True Positives (TPR): Sensitivity is another word for it. The chance that a real positive will show up in a test. So, here's how it's defined:

TPR=TP/TP+FN

The rate of false positives (FPR): It is the chance of a false alarm, which is when a positive result is given when the real value is negative. Where FP is the number of false positives and TN is the number of true negatives. (FP+TN is the total number of negatives.) Here's how it's defined:

FPR=FP/FP+TN

An ROC curve shows the relationship between TPR and FPR at different levels of classification. When the classification threshold is lowered, more items are marked as positive, which increases both the number of False Positives and the number of True Positives.

4.3 Outcome:

The Validation Summary F1-

Score, Precision, Recall, and ROC metrics for random forest algorithm were used as predictors. The results of testing our proposed model, which does a better job of predicting consumer behaviour using machine learning, will be used to show how these metrics work.

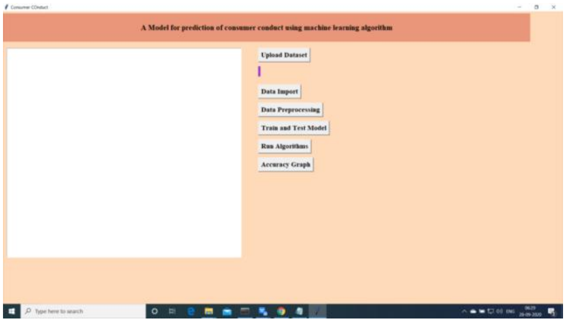


Fig.5

Now click on "Upload data "

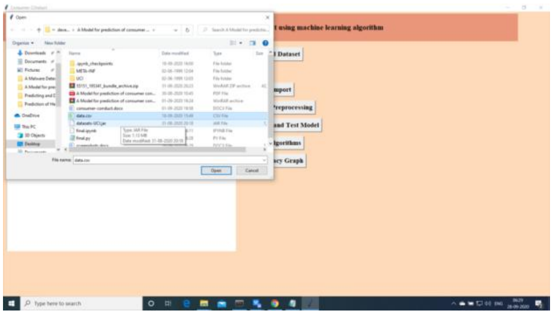
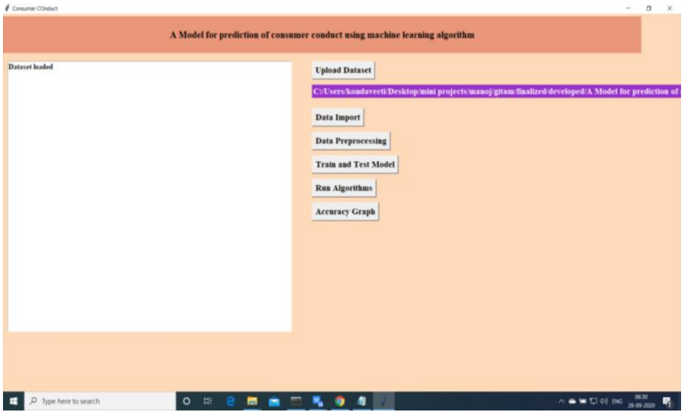


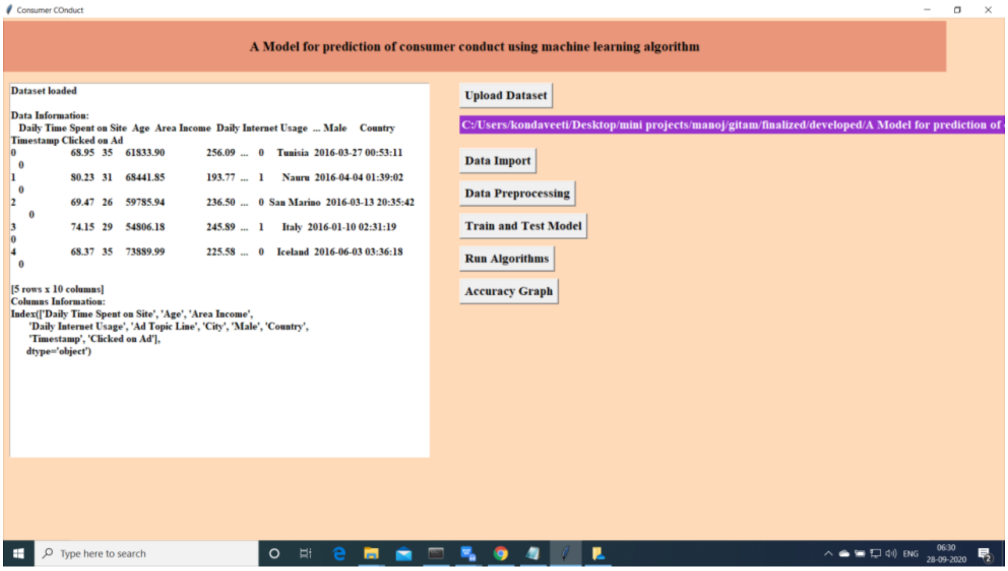
Fig.6



Dataset loaded

C:/Users/kondaveeti/Desktop/mini projects/manoj/gitam/finalized/developed/A Model for prediction of

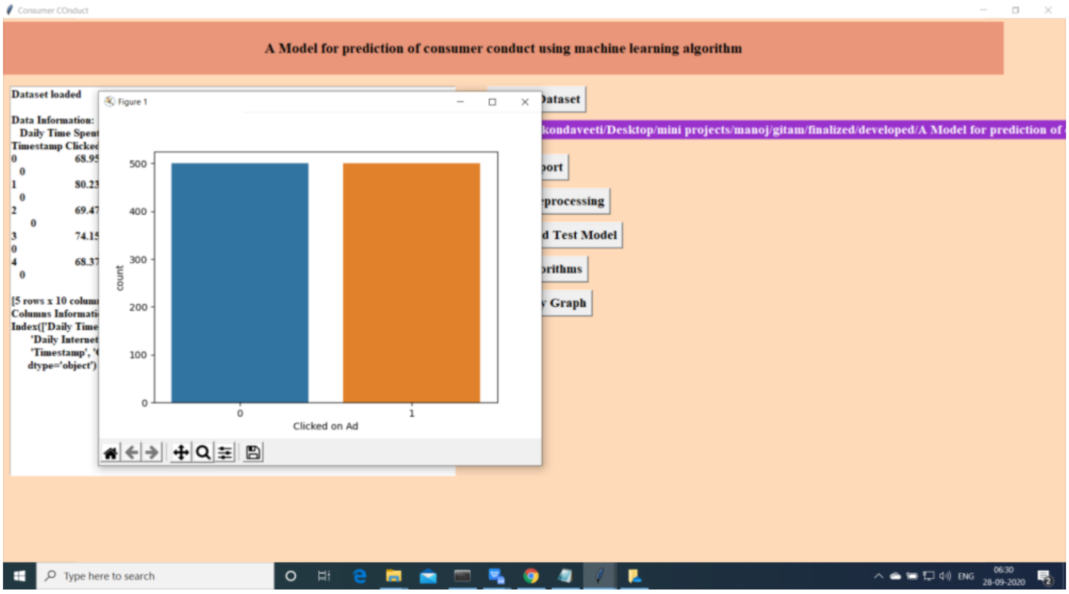
Fig.7. Import the data


Consumer COnduct
A Model for prediction of consumer conduct using machine learning algorithm
Upload Dataset
C:/Users/kondaveeti/Desktop/mini projects/manoj/gitam/finalized/developed/A Model for prediction of
Data Import
Data Preprocessing
Train and Test Model
Run Algorithms
Accuracy Graph
Type here to search
ENG
06:30
28-09-2020

```
Dataset loaded

Data Information:
  Daily Time Spent on Site  Age  Area Income  Daily Internet Usage ... Male    Country
Timestamp Clicked on Ad
0            68.95  35   61833.90          256.09 ...  0   Tunisia  2016-03-27 00:53:11
  0
1            80.23  31   68441.85          193.77 ...  1     Nauru  2016-04-04 01:39:02
  0
2            69.47  26   59785.94          236.50 ...  0 San Marino  2016-03-13 20:35:42
     0
3            74.15  29   54806.18          245.89 ...  1     Italy  2016-01-10 02:31:19
0
4            68.37  35   73889.99          225.58 ...  0   Iceland  2016-06-03 03:36:18
  0

[5 rows x 10 columns]
Columns Information:
Index(['Daily Time Spent on Site', 'Age', 'Area Income',
      'Daily Internet Usage', 'Ad Topic Line', 'City', 'Male', 'Country',
      'Timestamp', 'Clicked on Ad'],
     dtype='object')
```

Fig.8. Upload the data and read the basic data information will be shown on the figure

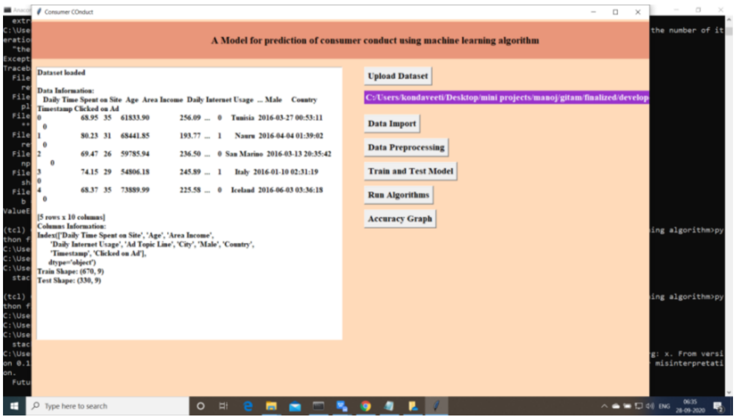
Data Preprocessing


Consumer COnduct
A Model for prediction of consumer conduct using machine learning algorithm
Figure 1
500
400
300
200
100
0
count
0
1
Clicked on Ad
Dataset
kondaveeti/Desktop/mini projects/manoj/gitam/finalized/developed/A Model for prediction of
port
processing
d Test Model
rithms
Graph
Type here to search
ENG
06:30
28-09-2020

```
Dataset loaded

Data Information:
  Daily Time Spent
Timestamp Clicked
0          68.95
  0
1          80.23
  0
2          69.47
     0
3          74.15
0
4          68.37
  0

[5 rows x 10 colum
Columns Informati
Index(['Daily Time
      'Daily Internet
      'Timestamp', '
     dtype='object')
```

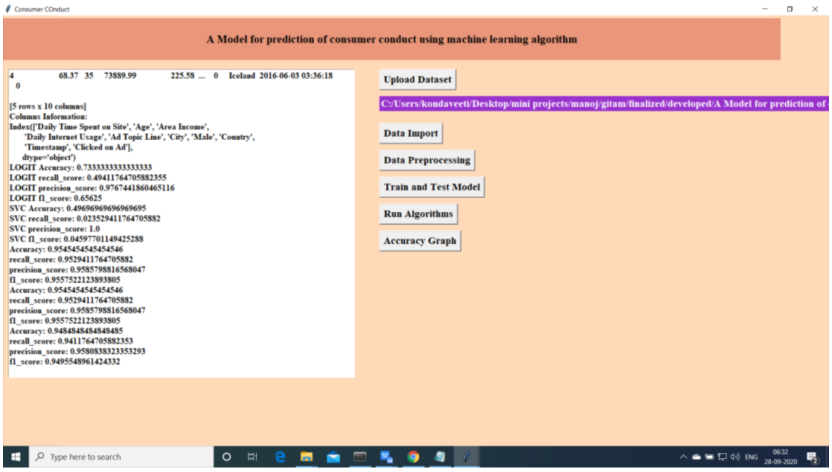
Fig.9. Preprocessing

Now click on “Train and Test model”. split the data into train and test and traain will be used for training and to tests the performances we are using test data


Consumer COnduct
A Model for prediction of consumer conduct using machine learning algorithm
Upload Dataset
C:/Users/kondaveeti/Desktop/mini projects/manoj/gitam/finalized/develop
Data Import
Data Preprocessing
Train and Test Model
Run Algorithms
Accuracy Graph
Type here to search

```
Dataset loaded

Data Information:
  Daily Time Spent on Site  Age  Area Income  Daily Internet Usage ... Male    Country
Timestamp Clicked on Ad
0                68.95  35   61833.90            256.09 ...  0    Tunisia  2016-03-27 00:53:11
  0
1                80.23  31   68441.85            193.77 ...  1     Nauru  2016-04-04 01:39:02
  0
2                69.47  26   59785.94            236.50 ...  0 San Marino  2016-03-13 20:35:42
    0
3                74.15  29   54806.18            245.89 ...  1     Italy  2016-01-10 02:31:19
0
4                68.37  35   73889.99            225.58 ...  0    Iceland  2016-06-03 03:36:18
  0

[5 rows x 10 columns]
Columns Information:
Index(['Daily Time Spent on Site', 'Age', 'Area Income',
      'Daily Internet Usage', 'Ad Topic Line', 'City', 'Male', 'Country',
      'Timestamp', 'Clicked on Ad'],
     dtype='object')
Train Shape: (670, 9)
Test Shape: (330, 9)
```

Fig.10. Now click on "Run Algorithms". Mentioned algorithms will be run on the data


Consumer COnduct
A Model for prediction of consumer conduct using machine learning algorithm
Upload Dataset
C:/Users/kondaveeti/Desktop/mini projects/manoj/gitam/finalized/developed/A Model for prediction of
Data Import
Data Preprocessing
Train and Test Model
Run Algorithms
Accuracy Graph
Type here to search

```
4                68.37  35   73889.99            225.58 ...  0    Iceland  2016-06-03 03:36:18
  0

[5 rows x 10 columns]
Columns Information:
Index(['Daily Time Spent on Site', 'Age', 'Area Income',
      'Daily Internet Usage', 'Ad Topic Line', 'City', 'Male', 'Country',
      'Timestamp', 'Clicked on Ad'],
     dtype='object')
LOGIT Accuracy: 0.7333333333333333
LOGIT recall_score: 0.49411764705882355
LOGIT precision_score: 0.9767441860465116
LOGIT f1_score: 0.65625
SVC Accuracy: 0.49696969696969695
SVC recall_score: 0.023529411764705882
SVC precision_score: 1.0
SVC f1_score: 0.04597701149425288
Accuracy: 0.9545454545454546
recall_score: 0.9529411764705882
precision_score: 0.9585798816568047
f1_score: 0.9557522123893805
Accuracy: 0.9545454545454546
recall_score: 0.9529411764705882
precision_score: 0.9585798816568047
f1_score: 0.9557522123893805
Accuracy: 0.9484848484848485
recall_score: 0.9411764705882353
precision_score: 0.9580838323353293
f1_score: 0.9495548961424332
```

Fig.11. Accuracy Comparison for all the models

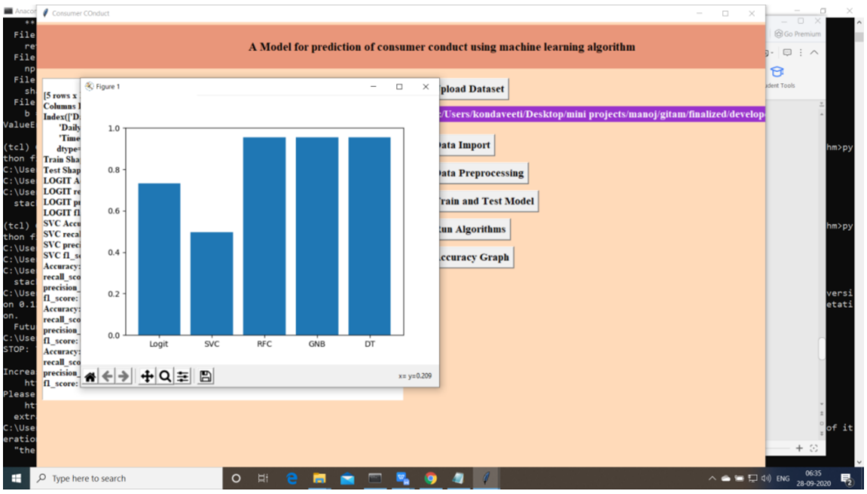


Fig.11

Extension Outcomes:

In this project author has used traditional machine learning algorithms such as SVM, Random Forest, logistic regression to predict consumer conduct such as consumer CLICKED on business linked or not while browsing.

In extension we have added advance machine learning algorithm called BaggingClassifier and compare its accuracy with traditional algorithms

A Bagging classifier is an ensemble meta-estimator that fits base classifiers each on random subsets of the original dataset and then aggregate their individual predictions (either by voting or by averaging) to form a final prediction. Such a meta-estimator can typically be used as a way to reduce the variance of a black-box estimator (e.g., a decision tree), by introducing randomization into its construction procedure and then making an ensemble out of it.

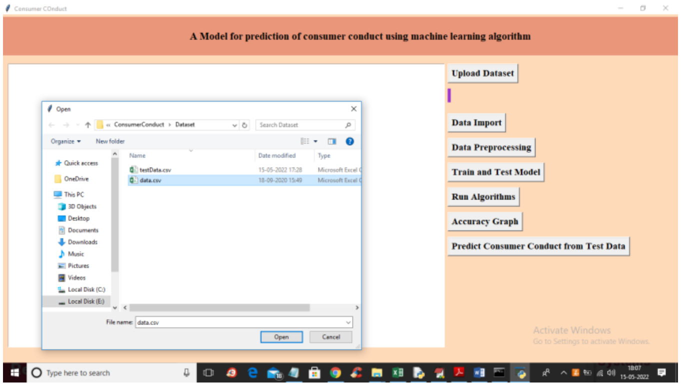
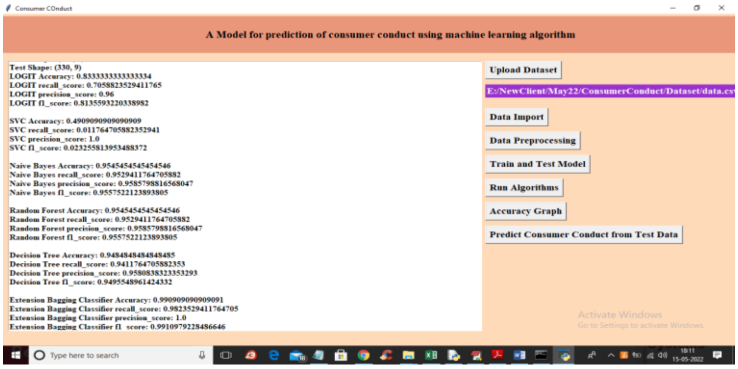


Fig.12. Upload dataset

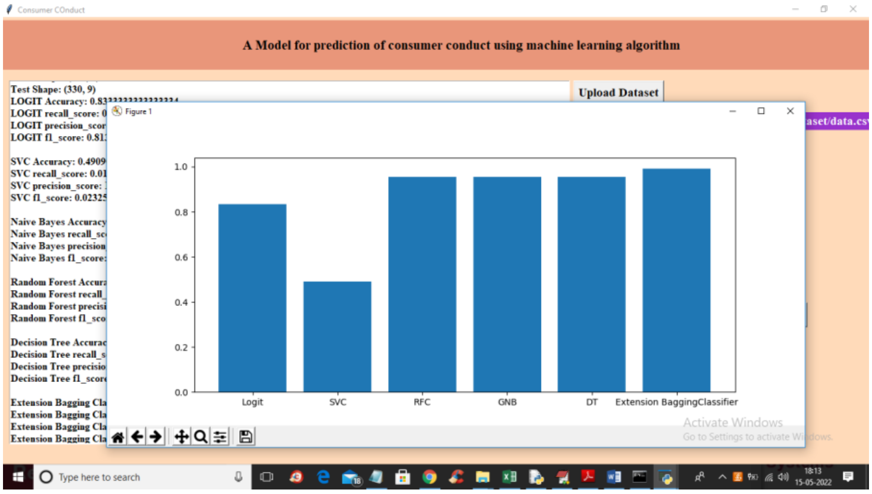
In above figure click on first button to upload dataset and then click remaining buttons and while training ML algorithms we can see output of extension BaggingClassifier like below figure


Consumer COnduct
A Model for prediction of consumer conduct using machine learning algorithm
Upload Dataset
E:/NewClient/May22/ConsumerConduct/Dataset/data.csv
Data Import
Data Preprocessing
Train and Test Model
Run Algorithms
Accuracy Graph
Predict Consumer Conduct from Test Data

```
Test Shape: (330, 9)
LOGIT Accuracy: 0.8333333333333334
LOGIT recall_score: 0.7058823529411765
LOGIT precision_score: 0.96
LOGIT f1_score: 0.8135593220338982

SVC Accuracy: 0.4909090909090909
SVC recall_score: 0.011764705882352941
SVC precision_score: 1.0
SVC f1_score: 0.023255813953488372

Naive Bayes Accuracy: 0.9545454545454546
Naive Bayes recall_score: 0.9529411764705882
Naive Bayes precision_score: 0.9585798816568047
Naive Bayes f1_score: 0.9557522123893805

Random Forest Accuracy: 0.9545454545454546
Random Forest recall_score: 0.9529411764705882
Random Forest precision_score: 0.9585798816568047
Random Forest f1_score: 0.9557522123893805

Decision Tree Accuracy: 0.9484848484848485
Decision Tree recall_score: 0.9411764705882353
Decision Tree precision_score: 0.9580838323353293
Decision Tree f1_score: 0.9495548961424332

Extension Bagging Classifier Accuracy: 0.990909090909091
Extension Bagging Classifier recall_score: 0.9823529411764705
Extension Bagging Classifier precision_score: 1.0
Extension Bagging Classifier f1_score: 0.9910979228486646
```

Fig.13. Accuracy

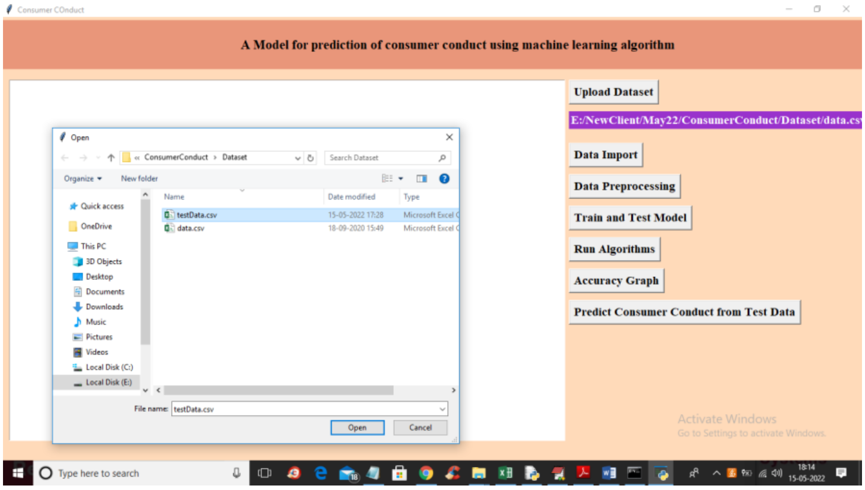
In above figure you can see accuracy of all traditional algorithms such as logistic regression, SVM, Naïve Bayes, Random Forest and Decision Tree but extension Bagging Classifier is giving high accuracy as 99% and now click on ‘Accuracy Graph button to get below graph


Consumer COnduct
A Model for prediction of consumer conduct using machine learning algorithm
Upload Dataset
aset/data.csv
Figure 1
1.0
0.8
0.6
0.4
0.2
0.0
Logit
SVC
RFC
GNB
DT
Extension BaggingClassifier
Activate Windows
Go to Settings to activate Windows
Type here to search
18:13
15-05-2022

```
Test Shape: (330, 9)
LOGIT Accuracy: 0.83333333333333334
LOGIT recall_score: 0
LOGIT precision_scor
LOGIT f1_score: 0.81

SVC Accuracy: 0.4909
SVC recall_score: 0.01
SVC precision_score: 
SVC f1_score: 0.02325

Naive Bayes Accuracy
Naive Bayes recall_sc
Naive Bayes precision
Naive Bayes f1_score:

Random Forest Accura
Random Forest recall_
Random Forest precisi
Random Forest f1_sco

Decision Tree Accurac
Decision Tree recall_s
Decision Tree precisio
Decision Tree f1_score

Extension Bagging Cla
Extension Bagging Cla
Extension Bagging Cla
Extension Bagging Cla
```

Fig.14. Algorithm name vs Accuracy

In above graph x-axis represents algorithm names and y-axis represents accuracy of all algorithms and in all algorithms extension got high accuracy and now close above graph and then click on ‘Predict Consumer Conduct from Test Data’ button to upload test data and then extension algorithm will predict consumer conduct like below figure



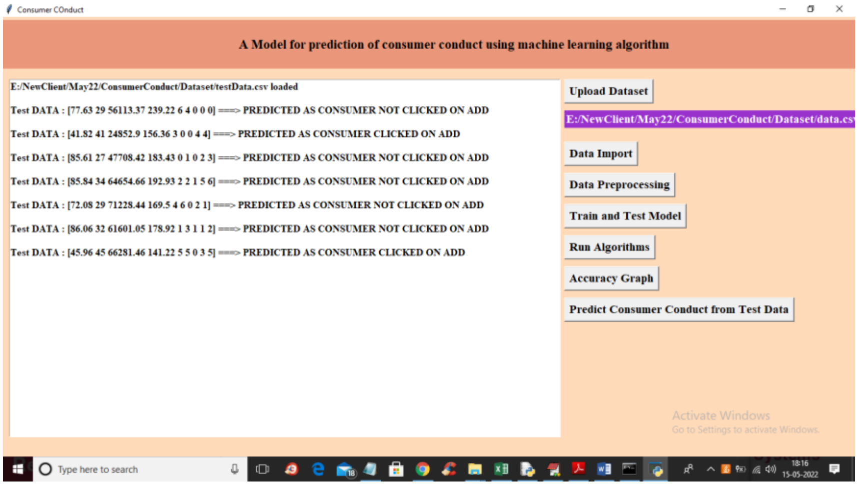
E:/NewClient/May22/ConsumerConduct/Dataset/data.csv

Name	Date modified	Type
testData.csv	15-05-2022 17:28	Microsoft Excel C
data.csv	18-09-2020 15:49	Microsoft Excel C

File name: testData.csv

Fig.15. Uploading ‘testData.csv’

In above figure selecting and uploading 'testData.csv' file and then click on 'Open' button to load dataset and get below output


Consumer COnduct
A Model for prediction of consumer conduct using machine learning algorithm
Upload Dataset
E:/NewClient/May22/ConsumerConduct/Dataset/data.csv
Data Import
Data Preprocessing
Train and Test Model
Run Algorithms
Accuracy Graph
Predict Consumer Conduct from Test Data

```
E:/NewClient/May22/ConsumerConduct/Dataset/testData.csv loaded
Test DATA : [77.63 29 56113.37 239.22 6 4 0 0 0] ===> PREDICTED AS CONSUMER NOT CLICKED ON ADD
Test DATA : [41.82 41 24852.9 156.36 3 0 0 4 4] ===> PREDICTED AS CONSUMER CLICKED ON ADD
Test DATA : [85.61 27 47708.42 183.43 0 1 0 2 3] ===> PREDICTED AS CONSUMER NOT CLICKED ON ADD
Test DATA : [85.84 34 64654.66 192.93 2 2 1 5 6] ===> PREDICTED AS CONSUMER NOT CLICKED ON ADD
Test DATA : [72.08 29 71228.44 169.5 4 6 0 2 1] ===> PREDICTED AS CONSUMER NOT CLICKED ON ADD
Test DATA : [86.06 32 61601.05 178.92 1 3 1 1 2] ===> PREDICTED AS CONSUMER NOT CLICKED ON ADD
Test DATA : [45.96 45 66281.46 141.22 5 5 0 3 5] ===> PREDICTED AS CONSUMER CLICKED ON ADD
```

Fig.16. Test data values

In above figure in square bracket we can see test data values and after square bracket we can see predicted values as 'consumer clicked on ADD or NOT'

CONCLUSION

The results of the observations show that the offered algorithm is more accurate than the Naive Bayes, AODE, and AODEsr algorithms. It takes a lot less time to learn than AODE and AODEsr. It is helpful to know that the procedure suggested in this paper is a complex one-dimensional model and is not the same as high-dimensional data.

Bibilography

1. Webb, G.I., Boughton, J., Wang, Z.: Not so naive bayes: Aggregating one-dependence estimators. Machine Learning 58, 5– 24 (2005) zbMATHCrossRefGoogle Scholar
2. Chen S., Martinez A.M., Webb G.I. (2014) Highly Scalable Attribute Selection for Averaged One-Dependence Estimators. In: Tseng V.S., Ho T.B., Zhou ZH., Chen A.L.P., Kao HY. (eds) Advances in Knowledge Discovery and Data Mining. PAKDD 2014. Lecture Notes in Computer Science, vol 8444. Springer, Cham.

3. Witten, I.H., Frank, E.: Data mining-Practical Machine Learning Tools and Techniques with Java Implementation. Morgan Kaufmann, San Francisco (2000), http://prdownloads.sourceforge.net/weka/datasets-UCI.jar Google Scholar
4. Pearl, J.: Probabilistic Reasoning in Intelligent Systems. Morgan Kaufmann, San Francisco (1988) Google Scholar
5. Merz, C., Murphy, P., Aha, D.: UCI repository of machine learning databases. In Dept of ICS, University of California, Irvine (1997), http://www.ics.uci.edu/mlearn/MLRepository.html
6. Langley, P., Sage, S.: Induction of selective Bayesian classifiers. In: Proceedings of the Tenth Conference on Uncertainty in Artificial Intelligence, pp. 339–406 (1994) Google Scholar
7. Kohavi, R.: Scaling Up the Accuracy of Naive-Bayes Classifiers: A Decision-Tree Hybrid. In: Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD 1996), pp. 202–207. AAAI Press, Menlo Park (1996) Google Scholar
8. Friedman, Geiger, Goldszmidt: Bayesian Network Classifiers. Machine Learning 29, 131–163 (1997) zbMATHCrossRefGoogle Scholar
9. Jiang L., Zhang H. (2006) Weightily Averaged One-Dependence Estimators. In: Yang Q., Webb G. (eds) PRICAI 2006: Trends in Artificial Intelligence. PRICAI 2006. Lecture Notes in Computer Science, vol 4099. Springer, Berlin, Heidelberg.
10. Chickering, D.M.: Learning Bayesian networks is NP-Complete. In: Fisher, D., Lenz, H. (eds.) Learning from Data: Artificial Intelligence and Statistics V, pp. 121–130. Springer, Heidelberg (1996) Google Scholar
11. Fei Zheng, Geoffrey I. Webb: Efficient Lazy Elimination for Averaged-One Dependence Estimators. In: Proceedings of the Twenty-third International Conference on Machine Learning (ICML 2006), 1113-1120, 2006