Redowan Shajib
CSCI 334
Instructor: Professor Bon Sy

Abstract:

This paper is an attempt to predict the engagement ratio (ER) of diabetes patients and provide recommendations for diet or physical exercise. It is expected that a better recommendation should help to increase ER among the patients. Given data of patients include different categories like cluster, gender, age, ethnicity, income, work-hours, health condition, education, motivation, attitude, intention, ownership, recommendation, and ER. A decision tree was used to find the recommendation and linear regression to find the ER. It is expected that the recommendations from this prediction will give a better ER in the next week.

Keywords: ER, recommendations, predictions, decision tree, entropy reduction, linear regression.

Methodology:

Different techniques like decision tree, mining association patterns, model discovery for pattern interference, SVD, Bayesian network, etc. were suggested for the prediction. A decision tree was used for both suggestion and ER prediction. A decision tree is a simple, powerful, and versatile machine learning algorithm that can perform both classifications and regression tasks. For a small dataset of 9 patients, it is expected to be a great algorithm. While using the decision tree to predict ER, I've noticed the fluctuation of ER in each week. As the standard deviation is pretty high for most of the weeks' ER data, linear regression was deemed to be a better approach. Linear regression finds the best-fit line to predict future data. It should be kept in mind that, both techniques carry the risk of oversimplification.

Data Mining Techniques:

Decision Tree

Click here for the Decision Tree code (Python)

Decision Tree Algorithm:
A decision tree is a flowchart-like tree structure where an internal node represents attributes like gender, age, ethnicity, etc. in our datasets. and each leaf node represents the outcome. The topmost node in a decision tree is known as the final root node (recommendation, estimated ER). This flowchart-like system allows you to quickly make decisions.

How does the Decision Tree algorithm work?

The basic idea behind any decision tree algorithm is as follows:

• Select the best attribute using Attribute Selection Measures (ASM) to split the records.
• Make this a decision node attribute and split the dataset into smaller subsets.
• Starts tree building by repeating this process recursively for each child until one of the conditions will match.
• All the tuples belong to the same attribute value.
• No more remaining attributes are available.
• There are no more instances.

Attribute selection measure:
The measures of attribute selection is a heuristic measure for selecting the splitting criterion in the best possible way to partition data. The most popular selection measures are Information Gain, Gain Ratio, and Gini Index.

Entropy: The entropy principle measures the impurity of the set of inputs In physics and mathematics, entropy referred to as the randomness or the impurity in the system. Here, it refers to the impurity in a group of examples.

Information gain: Data gain is the reduction in entropy Information gain computes the difference between entropy before the split and average entropy after sp

$$\text{Info(D)} = -\sum_{i=1}^{m} pi \log_2 pi$$

Where Pi is the probability that an arbitrary tuple in D belongs to class Ci.

$$\text{Info}_A(D)=\sum_{j=1}^{V}\frac{|Dj|}{|D|} \quad \text{X Info}(D_j)$$

$$\text{Gain}(A)=\text{Info}(D)-\text{Info}_A(D)$$

Here,
Info(D) is the average amount of information needed to identify the class label of a tuple in D.
|Dj|/|D| acts as the weight of the jth partition.
InfoA(D) is the expected information required to classify a tuple from D based on the partitioning by A.
The attribute A with the highest information gain, Gain(A), is chosen as the splitting attribute at node N ().

Gain Ratio:
Gaining data is skewed with many findings for the attribute It means it prefers the attribute with a large number of distinct values.

$$SplitInfo_A(D) = -\sum_{j=1}^{v}\frac{|D_j|}{|D|}\times \log_2\left(\frac{|D_j|}{|D|}\right)$$

|Dj|/|D| acts as the weight of the jth partition.
v is the number of discrete values in attribute **A**.

The gain ratio can be defined as

$$GainRatio(A) = \frac{Gain(A)}{SplitInfo_A(D)}$$

The attribute with the highest gain ratio is chosen as the splitting attribute.

Gini index:

Another decision tree algorithm CART (Classification and Regression Tree)

$$\text{Gini}(D)=1-\sum_{i=1}^{m} P_i^2$$

Where pi is the probability that a tuple in D belongs to class Ci. For each attribute, the Gini Index considers a binary break. You can compute a weighted sum of the impurity of each partition. If a binary split on attribute A partitions data D into D1 and D2, the Gini index of D is:

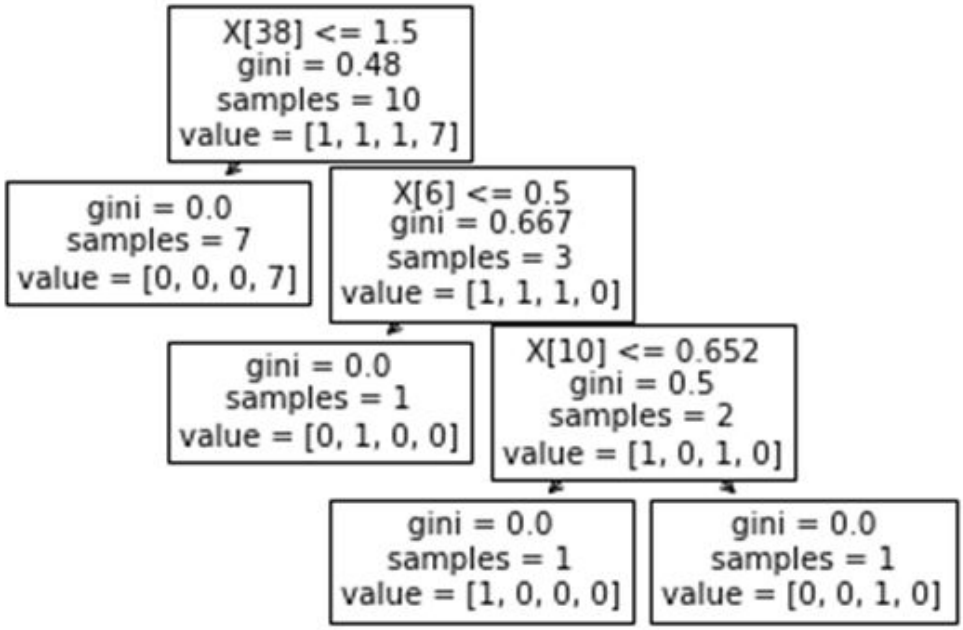
$$\text{Gini}_A(D)=\frac{|D1|}{|D|}\text{Gini}(D_1)+\frac{|D2|}{|D|}\text{Gini}(D_2)$$

The subset that gives the minimum Gini index for that chosen attribute is selected as a splitting attribute for a discrete-valued attribute. The strategy is to select each pair of adjacent values as a potential split-point and point with a smaller Gini index selected as a split-point in the case of continuous-valued attributes.

$$\Delta Gini(A) = Gini(D) - Gini_A(D).$$

The attribute with the minimum Gini index is chosen as the splitting attribute.

Visualizing the tree: Based on week 4 data -



Note: We observe the lowest Gini Index and hence it will be chosen as the root node for the decision tree.

•Accuracy:

Accuracy is one metric for evaluating classification models likewise Decision Tree. Informally, accuracy is the fraction of predictions in our model. Accuracy can be defined as:

$$Accuracy = \frac{Number\ of\ Features\ classified\ correctly}{Number\ of\ Features}$$

$$Misclassification\ Rate = \frac{Number\ of\ Features\ classified\ incorrectly}{Number\ of\ Features} = 1 - Accuracy$$

In the week-4 dataset, we can find 66.6354% accuracy for Recommendation and 64.4456% accuracy for Estimated ER. Accuracy comes out to 66.6354% and 64.4456% (66.6354 and 64.4456 correct predictions out of 100 total examples). That means our classifier is not doing a great job of identifying Recommendation and Estimated ER. So, in the future, we cannot use it for more data.

Linear Regression

The simple linear regression model is represented by the equation:

$y = \alpha + \beta X$

By mathematical convention, the two factors that are involved in simple linear regression analysis are designated X and y. The equation that describes how y is related to x is known as the regression model. Here in equation α is the y-intercept of the regression line and β is the slope.

The linear regression technique works on the following algorithm

Step 1: Take the values of variable Xi and Yi

Step 2: Calculate the average for variable Xi such that average is x= (X1 +X2 +……..+ Xi)/ Xi

Step 3: Calculate the average for variable Yi such that average is y= (Y1 +Y2 +……..+ Yi)/ Yi

Step 4: Calculate the value of regression coefficient β by substituting the values of Xi, Yi average of Xi and average of Yi in equation 2

Step 5: Calculate the value of another regression coefficients α by substituting the values of β (calculated in step 4), an average of Xi, and an average of Yi in equation 3

Step 6: Finally substitute the value of regression coefficients α and β in the equation Y= α + βX

Data collection:

Required data is chosen in two steps -

• Raw data collection: Provided in class using the spreadsheet. Click this link for the

spreadsheet.

• Data Pre-processing- This step makes the data ready to be used. Processes can include handling of noisy values, removal of redundant values, selection of proper attributes, etc.

Results:

Decision tree

For the decision tree, the recommendation and prediction for this week is -

Row number in the spreadsheet	Recommendation	Expected ER
4	**Phy**	0.69221
6	**Diet**	0.22608
7	**Diet**	0.87431
8	**Diet**	0.29189
9	**Diet**	0.52301
11	**Phy**	0.69123
12	**Phy**	0.69180

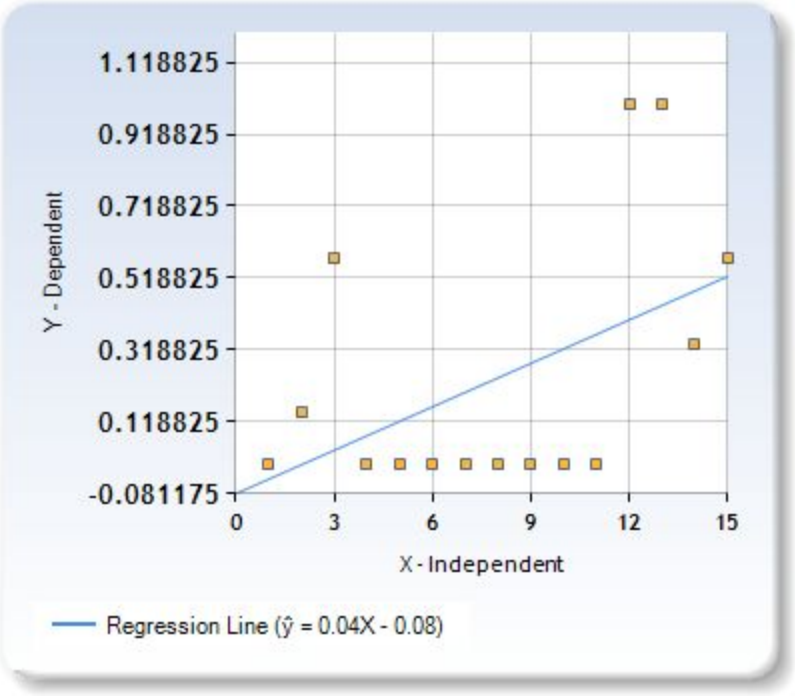
Linear Regression

Patient 1 (Row 4) ER: 0 , 0.142857, 0.571429, 0, 0, 0 ,0 ,0 ,0 ,0 ,0, 1, 1, 0.3333, 0.571429

For this data, the regression equation is -

$\hat{y} = 0.04031X - 0.08118$

Goodness to fit:
R square - 0.2646
Sy.x - 0.3288
P-value - 0.0498



Sum of X = 120
Sum of Y = 3.619
Mean X = 8
Mean Y = 0.2413
Sum of squares (SS_X) = 280
Sum of products (SP) = 11.2854

Regression Equation = $\hat{y} = bX + a$

$b = SP/SS_X = 11.29/280 = 0.04031$

$a = M_Y - bM_X = 0.24 - (0.04*8) = -0.08118$
$\hat{y} = 0.04031X - 0.08118$

The estimated ER for next week is: **0.5637**

Patient 2 (Row 6) ER: 0.285714,1, 0.470588, 0.9, 0.916667, 1, 0.9

For this data, the regression equation is -

$$\hat{y} = 0.08175X + 0.45483$$

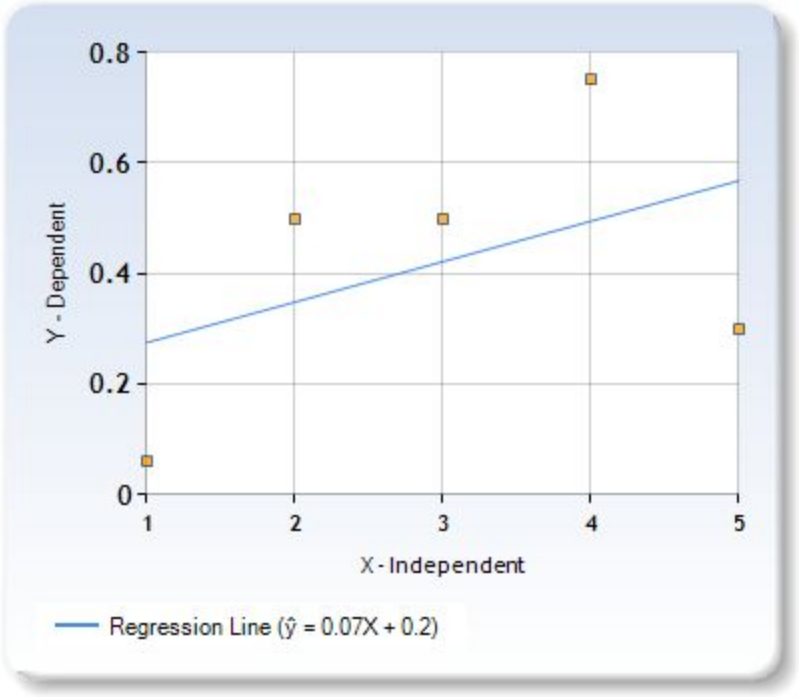
The estimated value becomes more than 1. So this value won't be used.

Patient 3 (Row 7) ER: 0.05882, 0.5, 0.5, 0.75, 0.3

For this data, the regression equation is -

$$\hat{y} = 0.07324X + 0.20204$$

Goodness to fit:
R square - 0.2012
Sy.x - 0.2664
P-value - 0.4486



The estimated ER for next week is: 0.6415

Patient 4(Row 8) ER : 0.25, 0 , 0.1

For this data, the regression equation is -

$$\hat{y} = -0.075X + 0.26667$$

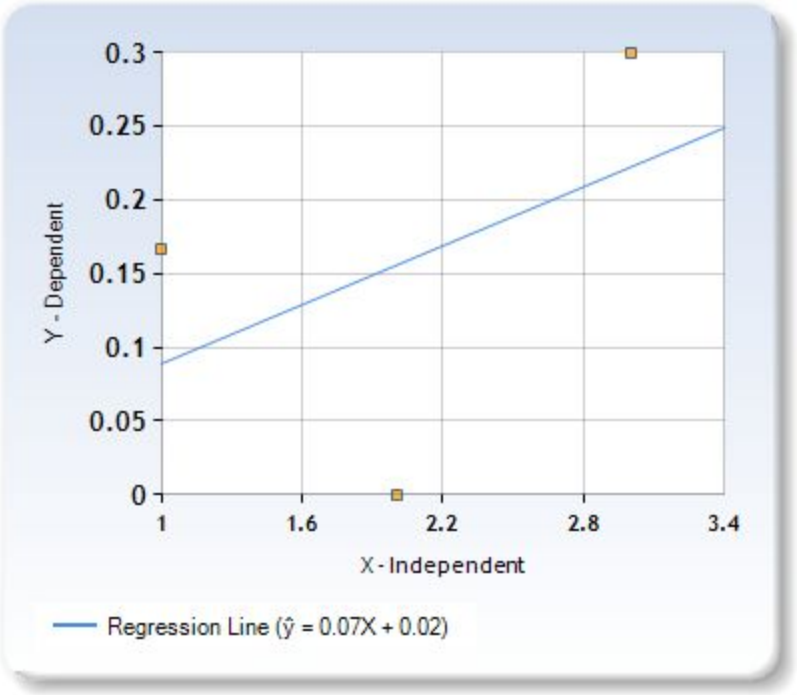
The estimated value becomes negative, so it won't be used.

Patient 5 (Row 9) ER: 0.1666667, 0, .3

For this data, the regression equation is -

$$\hat{y} = 0.06667X + 0.02223$$

Goodness to fit:
R square - 0.1967
Sy.x - 0.1905
P-value - 0.7074



The estimated ER for next week is: **0.2889**

Patient 6 (Row 11): Not enough data to run regression. Result of decision tree would be used.

Patient 7 (Row 12): Not enough data to run regression. Result of decision tree would be used.

FUTURE WORK

The efficiency of this prediction can be increased by implementing random forest or other learning algorithms as some of the data can't be analyzed with the current model. Accuracy comparison between different techniques can be practiced.