UNIVERSITY OF BENIN

CPE 593 ASSIGNMENT ON

MACHINE LEARNING TECHNIQUES

GROUP 5

S/N	NAME	MAT NO
1	EBU JACOB SUNDAY	ENG1503561
2	GODWIN MONICA OGHENERO	ENG1503570
3	GWAM ONYELUKACHUKWU NORA	ENG1508326
4	ADONNE ALEXANDER WENIEBI	ENG1503550
5	OPENE EKENEDINICHUKWU VICTORIA	ENG1503604
6	OMOWA FEMI JOEL	ENG1503602
7	ISAAC TAMUNO-ODUAM GAIUS	ENG1503578
8	OTUOLE UCHECHUKWU EMMANUEL	ENG1503608
9	IBEKWE THANKGOD ONYEDIKA	ENG1503571

MACHINE LEARNING

Machine learning is simply the application of artificial intelligence (AI) that provides systems the ability to automatically learn and improve from experience without being explicitly programmed. A term common to machine learning is Linear Regression. This section describes linear regression as a supervised form of Machine learning, other approach to machine learning is the unsupervised and semi-supervised form of learning.

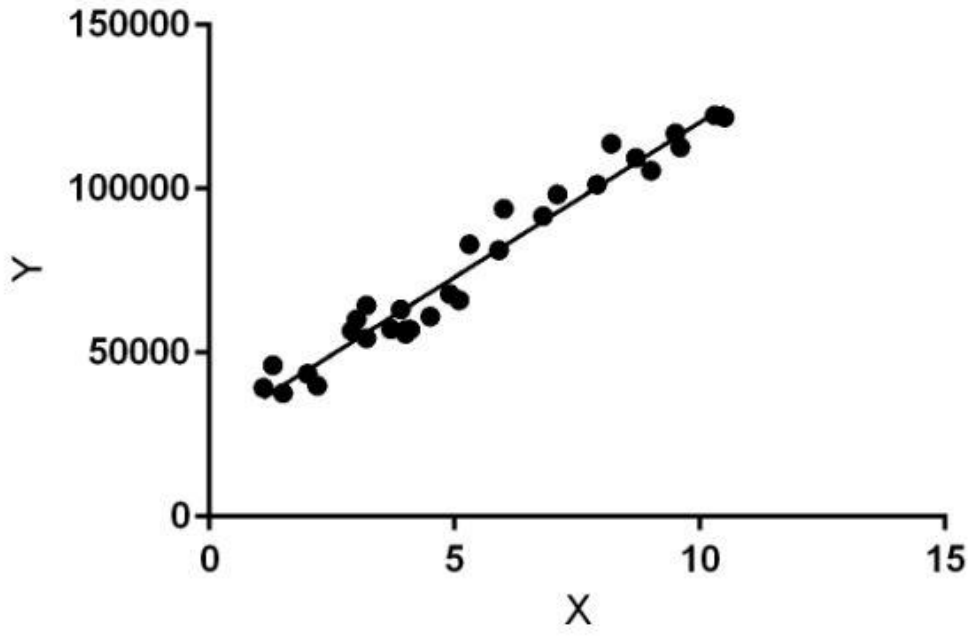
The Supervised machine learning algorithms is simply to apply what has been learned in the past to new set of data using labeled examples to predict future events.

MACHINE LEARNING TECHNIQUES

- LINEAR REGRESSION

Linear Regression is a machine learning algorithm based on supervised learning. It performs a regression task. Regression is a method of modeling a target value based on independent predictors. This method is mostly used for forecasting and finding out cause and effect relationship between variables. Regression techniques mostly differ based on the number of independent variables and the type of relationship between the independent and dependent variables.

Linear regression performs the task to predict a dependent variable value (y) based on a given independent variable (x). So, this regression technique finds out a linear relationship between x (input) and y(output). Hence, the name is Linear Regression.



In the figure above, X (input) is the work experience and Y (output) is the salary of a person. The regression line is the best fit line for our model.

Hypothesis function for Linear Regression:

$$y = \theta_1 + \theta_2.x$$

While training the model we are given:x: input training data (univariate – one input variable(parameter))y: labels to data (supervised learning)

When training the model – it fits the best line to predict the value of y for a given value of x. The model gets the best regression fit line by finding the best θ_1 and θ_2 values.θ_1: interceptθ_2: coefficient of x

Once we find the best θ_1 and θ_2 values, we get the best fit line. So when we are finally using our model for prediction, it will predict the value of y for the input value of x.

The principal advantage of linear regression is its simplicity, interpretability, scientific acceptance, and widespread availability. Linear regression is the first method to use for many problems. Looking at its interpretation using a cartesian coordinate x and y it is easily readable and interpreted as basic mathematical principles are applied. This form which it presents itself makes it scientifically acceptable as a analysis and prediction model.

Disadvantage of linear Regression is that many real-world phenomena simply do not correspond to the assumptions of a linear model; in these cases, it is difficult or impossible to produce useful results with linear regression. It is sometimes not practical for real life situations because of its over simplicity approach to real word problems.

- KNN TECHNIQUES

K-Nearest Neighbor is one of the simplest Machine Learning algorithms based on Supervised Learning technique, this algorithm assumes the similarity between the new case/data and available cases and put the new case into the category that is most similar to the available categories. It also stores all the available data and classifies a new data point based on the similarity. This means when new data appears then it can be easily classified into a well suite category by using K- NN algorithm.

K-NN algorithm can be used for Regression as well as for Classification but mostly it is used for the Classification problems, it is a non-parametric algorithm, which means it does not

make any assumption on underlying data. It is also called a lazy learner algorithm because it does not learn from the training set immediately instead it stores the dataset and at the time of classification, it performs an action on the dataset.

KNN algorithm at the training phase just stores the dataset and when it gets new data, then it classifies that data into a category that is much similar to the new data.

Example: Suppose, we have an image of a creature that looks similar to cat and dog, but we want to know either it is a cat or dog. So for this identification, we can use the KNN algorithm, as it works on a similarity measure. Our KNN model will find the similar features of the new data set to the cats and dogs images and based on the most similar features it will put it in either cat or dog category.

Step-1: Select the number K of the neighbors

Step-2: Calculate the Euclidean distance of K number of neighbors

Step-3: Take the K nearest neighbors as per the calculated Euclidean distance.

Step-4: Among these k neighbors, count the number of the data points in each category.

Step-5: Assign the new data points to that category for which the number of the neighbor is maximum.

Step-6: Our model is ready.

- K-MEANS OF ALGORITHM

We are given a data set of items, with certain features, and values for these features (like a vector). The task is to categorize those items into groups. To achieve this, we will use the K-means algorithm; an unsupervised learning algorithm.
K-means uses an iterative refinement method to produce its final clustering based on
the number of clusters defined by the user (represented by the variable K) and the dataset. For example, if you set K equal to 3 then your dataset will be grouped in 3 clusters, if you set K equal to 4 you will group the data in 4 clusters, and so on. K-means starts off with arbitrarily chosen data points as proposed means of the data groups, and iteratively recalculates new means in order to converge to a final clustering of the data points.
But how does the algorithm decide how to group the data if you are just providing a value (K)? When you define the value of K you are actually telling the algorithm how many means or centroids you want (if you set K=3 you create 3 means or centroids, which accounts for 3 clusters). A centroid is a data point that represents the center of the cluster (the mean), and it might not necessarily be a member of the dataset.
This is how the algorithm works:
(1) K centroids are created randomly (based on the predefined value of K)
(2) K-means allocates every data point in the dataset to the nearest centroid (minimizing Euclidean distances between them), meaning that a data point is considered to be in a particular cluster if it is closer to that cluster's centroid

than any other centroid
(3) Then K-means recalculates the centroids by taking the mean of all data points assigned to that centroid's cluster, hence reducing the total intra-cluster variance in relation to the previous step. The "means" in the K-means refers to averaging the data and finding the new centroid
The algorithm iterates between steps 2 and 3 until some criteria is met (e.g. the sum of distances between the data points and their corresponding centroid is minimized, a maximum number of iterations is reached, no changes in centroids value or no data points change clusters)
The initial result of running this algorithm may not be the best possible outcome and rerunning it with different randomized starting centroids might provide a better performance (different initial objects may produce different clustering results). For this reason, it's a common practice to run the algorithm multiple times with different starting points and evaluate different initiation methods. But another question arises: how do you know the correct value of K, or how many centroids to create?
There is no universal answer for this, and although the optimal number of centroids or clusters is not known a priori, different approaches exist to try to estimate it. One commonly used approach is testing different numbers of clusters and measure the resulting sum of squared errors, choosing the K value at which an increase will cause a very small decrease in the error sum, while a decrease will sharply increase the error sum. This point that defines the optimal number of clusters is known as the "elbow point", and can be used as a visual measure to find the best pick for the value of K.
K-means is a must-have in your data science toolkit, and there are several reasons for this. First of all it's easy to implement and brings an efficient performance. After all, you need to define just one parameter (the value of K) to see the results. It is also fast and works really well with large datasets, making it capable of dealing with the current huge volumes of data. It's so flexible that it can be used with pretty much any datatype and its results are easy to interpret and more explainable than other algorithms. Furthermore, the algorithm is so popular that you may find use cases and implementations in almost any discipline.

- SUPPORT VECTOR MACHINE

Support Vector Machine or SVM is one of the most popular Supervised Learning algorithms, which is used for Classification as well as Regression problems. However, primarily, it is used for Classification problems in Machine Learning.
The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane.
SVM chooses the extreme points/vectors that help in creating the hyperplane. These extreme cases are called as support vectors, and hence algorithm is termed as Support Vector Machine.

Hyperplanes are decision boundaries that help classify the data points. Data points falling on either side of the hyperplane can be attributed to different classes. Also, the dimension of the hyperplane depends upon the number of features. If the number of input features is 2, then the hyperplane is just a line. If the number of input features is 3, then the hyperplane becomes a two-dimensional plane. It becomes difficult to imagine when the number of features exceeds 3.

Support vectors are data points that are closer to the hyperplane and influence the position and orientation of the hyperplane. Using these support vectors, we maximize the margin of the classifier. Deleting the support vectors will change the position of the hyperplane. These are the points that help us build our SVM.
In SVM, we take the output of the linear function and if that output is greater than 1, we identify it with one class and if the output is -1, we identify is with another class. Since the threshold values are changed to 1 and -1 in SVM, we obtain this reinforcement range of values ([-1,1]) which acts as margin.

- DIMENSIONALITY REDUCTION

In machine learning classification problems, there are often too many factors on the basis of which the final classification is done. These factors are basically variables called features. The higher the number of features, the harder it gets to visualize the training set and then work on it. Sometimes, most of these features are correlated, and hence redundant. This is where dimensionality reduction algorithms come into play. Dimensionality reduction is the process of reducing the number of random variables under consideration, by obtaining a set of principal variables. It can be divided into feature selection and feature extraction.

Feature selection is the process of identifying and selecting relevant features for your sample. Feature engineering is manually generating new features from existing features, by applying some transformation or performing some operation on them. Feature selection can be done either manually or programmatically. For example, consider you are trying to build a model which predicts people's weights and you have collected a large corpus of data which describes each person quite thoroughly

Methods of Dimensionality Reduction

The various methods used for dimensionality reduction include:

- Principal Component Analysis (PCA)
- Linear Discriminant Analysis (LDA)
- Factor Analysis

Dimensionality reduction may be both linear or non-linear, depending upon the method used. The prime linear method, called Principal Component Analysis, or PCA, is discussed below

Principal Component Analysis: This method was introduced by Karl Pearson. It works on a condition that while the data in a higher dimensional space is mapped to data in a lower dimension space, the variance of the data in the lower dimensional space should be maximum.

Factor Analysis: A technique that is used to reduce a large number of variables into fewer numbers of factors. The values of observed data are expressed as functions of a number of possible causes in order to find which are the most important. The observations are assumed to

be caused by a linear transformation of lower dimensional latent factors and added Gaussian noise.

LDA (Linear Discriminant Analysis): Its projects data in a way that the class separability is maximized. Examples from same class are put closely together by the projection. Examples from different classes are placed far apart by the projection

Advantages of Dimensionality Reduction

- It helps in data compression, and hence reduced storage space.
- It reduces computation time.
- It also helps remove redundant features, if any.

Disadvantages of Dimensionality Reduction

- It may lead to some amount of data loss.
- PCA tends to find linear correlations between variables, which is sometimes undesirable.
- PCA fails in cases where mean and covariance are not enough to define datasets.
- We may not know how many principal components to keep- in practice, some thumb rules are applied.

- DECISION TREE LEARNING

Decision tree learning is one of the predictive modeling approaches used
in statistics, data mining and machine learning. It uses a decision tree (as a predictive model) to go from observations about an item (represented in the branches) to conclusions about the item's target value (represented in the leaves)

You have a question, usually a yes or no (binary; 2 options) question with two branches (yes and no) leading out of the tree. You can get more options than 2, but for this article, we're only using 2 options.

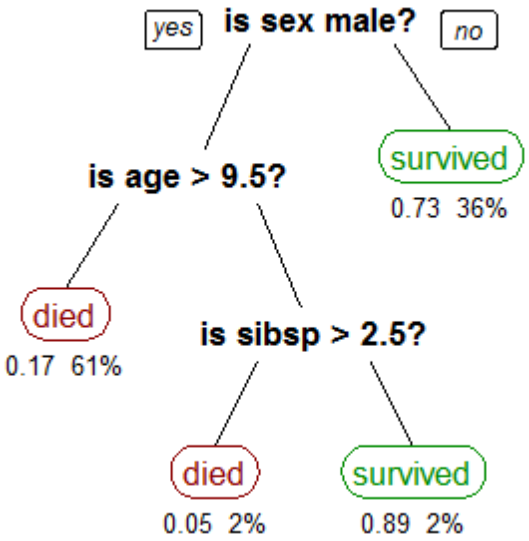
Trees are weird in computer science. Instead of growing from a root upwards, they grow downwards. Think of it as an upside down tree.

The top-most item, in this example, "Am I hungry?" is called
the *root*. It's where everything starts from. *Branches* are what we call each line. A *leaf* is everything that isn't the root or a branch.

Trees are important in machine learning as not only do they let us visualize an algorithm, but they are a type of machine learning. Take this algorithm as an example. This algorithm predicts the probability that a passenger will survive on the Titanic.

"*sibsp*" is the number of spouses or siblings aboard the ship. The figures under each leaf show the probability of survival.

With machine learning trees, the bold text is a condition. It's not data, it's a question. The branches are still called branches. The leaves are "*decisions*". The tree has decided whether someone would have survived or died.



Advantage of Decision tree in Machine learning

One big advantage of the decision tree model is its

1. transparent nature: Unlike other decision-making models, the decision tree makes explicit all possible alternatives and traces each alternative to its conclusion in a single view, allowing for easy comparison among the various alternatives. The use of separate nodes to denote user defined decisions, uncertainties, and end of process lends further clarity and transparency to the decision-making process.

2. Comprehensive Nature:

The decision tree is the best predictive model as it allows for a comprehensive analysis of the consequences of each possible decision, such as what the decision leads to, whether it ends in uncertainty or a definite conclusion, or whether it leads to new issues for which the process needs repetition.

3.Ease of Use

Decision trees also score in ease of use. The decision tree provides a graphical illustration of the problem and various alternatives in a simple and easy to understand format that requires no explanation.

Disadvantage:

1. A small change in the data can cause a large change in the structure of the decision tree causing instability.

2. For a Decision tree sometimes calculation can go far more complex compared to other algorithms.

3. Decision tree often involves higher time to train the model.
4. Decision tree training is relatively expensive as complexity and time taken is more.
5. Decision Tree algorithm is inadequate for applying regression and predicting continuous values.