**HiLCoE**

**School of computer science and technology**

**Data Mining Assignment**

**Group Names**

**1. Abubeker Nasir**

**2. Afomiya Belay**

**3. Mahlet Zelalem**

**4. Sami Mursela**

**5. Welid Abdulhafiz**

**Submitted to – Eyob N. (PhD)**

**Due date- 27/05/2022**

**Introduction**

**Stroke**

Stroke is a disease that affects the arteries leading to and within the brain. It is the No. 5 cause of death and a leading cause of disability in the United States.

A stroke occurs when a blood vessel that carries oxygen and nutrients to the brain is either blocked by a clot or bursts (or ruptures). When that happens, part of the brain cannot get the blood (and oxygen) it needs, so it and brain cells die.

​​What are the types of stroke?

Stroke can be caused either by a clot obstructing the flow of blood to the brain (called an [ischemic stroke](https://www.stroke.org/en/about-stroke/types-of-stroke/ischemic-stroke-clots)) or by a blood vessel rupturing and preventing blood flow to the brain (called a [hemorrhagic stroke](https://www.stroke.org/en/about-stroke/types-of-stroke/hemorrhagic-strokes-bleeds)). A TIA ([transient ischemic attack](https://www.stroke.org/en/about-stroke/types-of-stroke/tia-transient-ischemic-attack)), or "mini stroke", is caused by a temporary clot.

## What are the effects of stroke?

The brain is an extremely complex organ that controls various body functions. If a stroke occurs and blood flow can't reach the region that controls a particular body function, that part of the body won't work as it should.

The main problem of stroke is that it is unpredictable. Although strokes appear to hit victims indiscriminately, they often occur when precursory signs go unheeded. For example, individuals who have untreated hypertension (high blood pressure) are at higher risk of having a stroke.

**Objective**

The objective of this project is to predict if someone is at risk of having stroke using two classification algorithms.

**Methodology**

To implement the data mining task we used python and Jupyter notebook as an IDE. This is because from the research we made, the jupyter notebook is very suitable for machine learning. The data mining task we have used for this project is classification. Classification is a data mining function that assigns items in a collection to target categories or classes. The goal of classification is to accurately predict the target class for each case in the data. A classification task begins with a data set in which the class assignments are known. Classifications are discrete and do not imply order. Continuous, floating-point values would indicate a numerical, rather than a categorical, target. A predictive model with a numerical target uses a regression algorithm, not a classification algorithm.

In the model build (training) process, a classification algorithm finds relationships between the values of the predictors and the values of the target. Different classification algorithms use different techniques for finding relationships. These relationships are summarized in a model, which can then be applied to a different data set in which the class assignments are unknown.

We have chosen two algorithms for classification:

1. Decision tree classifier
2. KNN classifier

**Decision tree classifier**

Decision Tree is a Supervised learning technique that can be used for both classification and Regression problems, but mostly it is preferred for solving Classification problems. It is a tree-structured classifier, where internal nodes represent the features of a dataset, branches represent the decision rules and each leaf node represents the outcome.

**KNN classifier**

The k-nearest neighbors algorithm, also known as KNN or k-NN, is a non-parametric, supervised learning classifier, which uses proximity to make classifications or predictions about the grouping of an individual data point. While it can be used for either regression or classification problems, it is typically used as a classification algorithm, working off the assumption that similar points can be found near one another.

**Review related works**

We have reviewed two international works since we did not find local works on this particular case. We have attached both links below in the reference.

The first work we reviewed was done by hindawi. It is titled “**Stroke Disease Detection and Prediction Using Robust Learning Approaches”**

The study approaches the problem by using methods like LR, DT classification, RF classification, and voting classifier. After creating four alternative models, the accuracy measures, namely accuracy score, precision score, recall score, and F1 score are used to compare them.

##### **Proposed Algorithms**

The most common disease identified in the medical field is stroke, which is on the rise year after year. Using the publicly accessible stroke prediction dataset, the study measured four commonly used machine learning methods for predicting brain stroke recurrence, which are as follows:

(i)Random forest

(ii)Decision tree

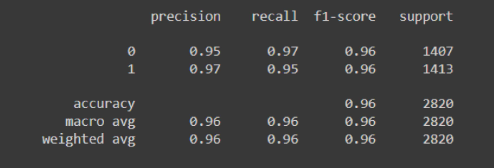
(iii)Voting classifier

(iv)Logistic regression

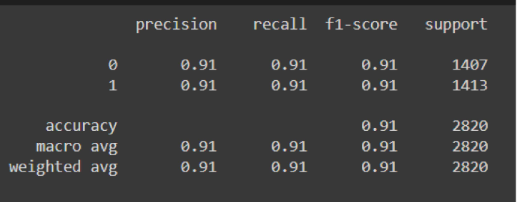
##### **Evaluation of the Model**

We have only viewed two of the algorithms evaluations

**Random Forest**

****

###### **Voting Classifier**

**We can conclude that the random forest has a better performance and accuracy compared to voting classifier**

###### The second work we reviewed was done by hindawi. It is titled “**Stroke Prediction Analysis using Machine Learning Classifiers and Feature Technique”** it was done by Uttara University Bangladesh the study states that “Machine learners have various applications expanding within the study of bioinformatics, a subfield of artificial intelligence which includes improving calculations to discover how projections are dependent on information.” “This paper utilized various machine learning (ML) models containing Naïve Bayes, Random Forest, Ada Boost Algorithm. Among them, the Random Forest model out performs the best accuracy. So Random forest model is described here.”

###### from the above table we can conclude that random forest has a better performance than the other algorithms

###### 

**Data Preparation**

First and foremost we tried to find different datasets that are suitable for this particular project. We looked into websites provided by the instructor, this are:

http://www.kdnuggets.com/datasets/

<https://favtutor.com/blogs/data-mining-projects>

http://www.inf.ed.ac.uk/teaching/courses/dme/html/datasets0405.html

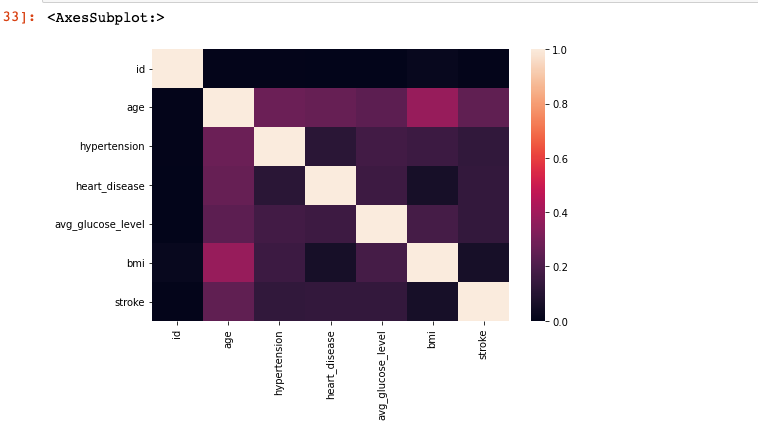
We found the stroke dataset in the first website. We chose this particular dataset because it has over 5000 data instances.

We then pre processed it by looking for null values, outlier values and noise. Then we cleaned them using different techniques.

The below code is to deal with outliers

****

After that we checked the heatmap of the dataset. A heatmap contains values representing various shades of the same color for each value to be plotted. Usually the darker shades of the chart represent higher values than the lighter shade. For a very different value a completely different color can also be used.



After that we could say that the data is clean. But we still need to differentiate the class label from the attributes. Obviously the class label was “​​Stroke” column. It had two classes, “1” and “0” meaning yes stroke and no stroke. After that we split the dataset as training dataset and test dataset. We did this using train\_test\_split library in sklearn. 

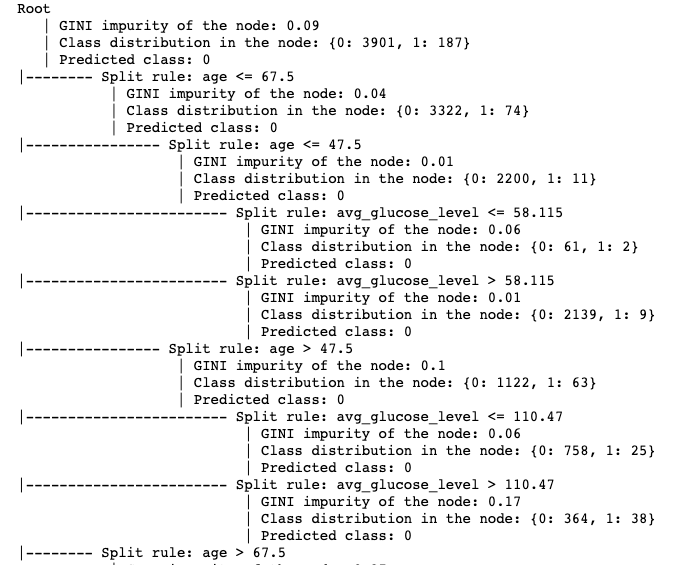
**Experimental setup** After the data has been cleaned we need to build the algorithm to do the data mining tasks.

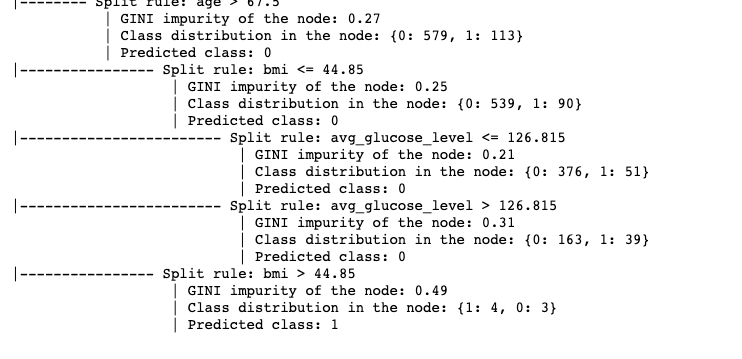
**Decision tree classifier** To write the algorithm for decision tree we looked into many references. There are lots of options we can use for decision tree algorithm like:

* ID3 (Iterative Dichotomiser 3)
* C4.5 (successor of ID3)
* CART (Classification And Regression Tree)
* Chi-square automatic interaction detection (CHAID). Performs multi-level splits when computing classification trees.
* MARS: extends decision trees to handle numerical data better.
* Conditional Inference Trees. Statistics-based approach that uses non-parametric tests as splitting criteria, corrected for multiple testing to avoid overfitting. This approach results in unbiased predictor selection and does not require pruning.

ID3 and CART were invented independently at around the same time (between 1970 and 1980), yet follow a similar approach for learning a decision tree from training tuples.

For this project we used the CART algorithm.The representation for the CART model is a binary tree.





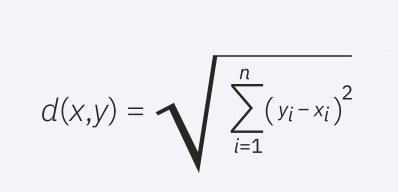
CART use Gini Impurity as the criterion to split node, not Information Gain. CART supports numerical target variables, which enables itself to become a Regression Tree that predicts continuous values.

1. **KNN classifier**

The goal of the k-nearest neighbor algorithm is to identify the nearest neighbors of a given query point, so that we can assign a class label to that point. In order to do this, KNN has a few requirements:

**Determine your distance metrics**

We used Euclidean distance for the distance. This is the most commonly used distance measure, and it is limited to real-valued vectors. Using the below formula, it measures a straight line between the query point and the other point being measured.

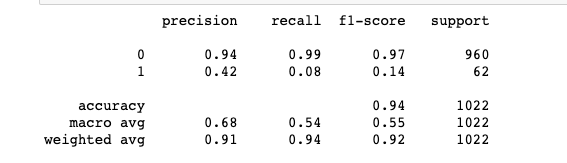


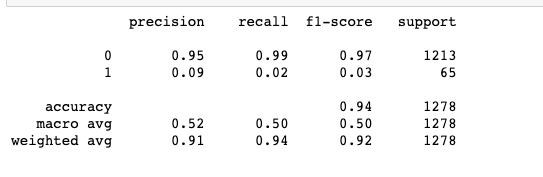
We also defined functions to calculate the nearest neighbor and to predict.

**Summary of experimental result & findings of the study**

To determine the performance of the algorithms we did a confusion matrix for both the algorithms. The first thing we did for both was use the model to predict the test data that we reserved earlier. Then store that in a variable and check this data with the original class and we then can build the confusion matrix.

The first image below is for the decision tree and the second one is for the KNN





When we compare the Two algorithms we can conclude that the KNN algorithm has a better precision than the decision tree. It also has better support.

**Concluding remarks**

We have created a 94% accurate classification model. We can also conclude that in a very small significance KNN is better than the decision tree in this particular case. We also concluded that to get better accuracy we need a lot more dataset.

**References**

* [**https://machinelearningmastery.com**](https://machinelearningmastery.com/classification-and-regression-trees-for-machine-learning/)
* [**https://www.tutorialspoint.com**](https://www.tutorialspoint.com/python_data_science/python_heat_maps.htm)
* [**https://www.askpython.com/**](https://www.askpython.com/)
* [**http://www.inf.ed.ac.uk/**](http://www.inf.ed.ac.uk/)
* [**https://www.familiprix.com/**](https://www.familiprix.com/)
* <https://www.indeed.com/>
* <https://docs.oracle.com/>
* <https://www.javatpoint.com/>
* <https://www.ibm.com/>
* <https://www.hindawi.com/journals/jhe/2021/7633381/>
* <http://ejournal.radenintan.ac.id/index.php/IJECS/article/view/10393/pdf>