PREDICT THE CLASS OF UNKNOWN PATIENT TO FIND A PROPER DRUG FOR A NEW PATIENT USING DECISION TREE ALGORTITHM

STREAM: M-TECH IN INFORMATION SECURITY

BY:

RITAM GHOSH

SAYANTA HARH

TANISHA GHOSHAL

QUAZI AHMED SAQUIB

ACKNOWLEDGEMENT

Completing this project made us even more knowledgeable in the field of Machine Learning and its applications. This project was very close to real life working projects and it vastly cleared our concepts in the field of classification problems. For this we are really grateful to our mentor Dr. Somdatta Chakrabortty for her immense help and support which helped us to successfully complete this project. Without her help this project could not have been completed successfully. We also thank each other for all the dedication everyone showed for completing this project. Hoping to get even more interesting projects in the future which will without a doubt will enhance our knowledge in this domain.

ABSTRACT

*Classification and Regression Trees or CART models, popularly called decision trees are an important algorithm to solve classification and regression modelling problems. It is one of the most widely used and practical models for supervised learning. Decision Tree models are non-parametric and discriminative. Decision Trees are simple to understand, interpret and visualize. They can easily handle both numerical and categorical and multi-output problems. It implicitly performs feature selection and thus requires relatively little effort from users for data preparation. Decision Trees have a test time complexity of O(depth) and train time complexity of O(n*log(n)*d). We have experimentally tested our proposed algorithm in JUPYTER NOTEBOOK, using a standard pharmaceutical dataset and trained a CART model to prescribe one of several drugs to an unknown patient.*

TABLE OF CONTENTS

INTRODUCTION

CLASSIFICATION IN MACHINE LEARNING:

Classification in machine learning and statistics is a supervised learning approach in which the computer program learns from the data given to it and makes new observations or classifications.

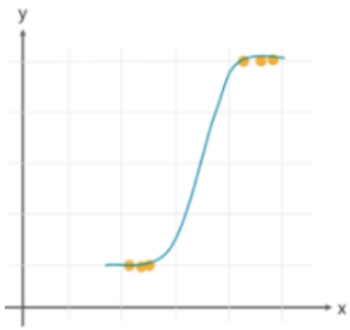
Classification is a process of ordering a given arrangement of data into classes. It can be performed on both structured and unstructured information. The process begins with predicting the class of given data points. The classes are regularly alluded to as target, name or classifications.

The classification predictive modelling is the task of approximating the mapping function from input variables to discrete output variables. The principle objective is to distinguish which class/classification the new information will fall into.

In machine learning, **classification** is a supervised learning concept which fundamentally orders a set of data into classes. The most common classification problems are – speech recognition, face detection, handwriting recognition, document classification, etc. It can be either a binary classification problem or a multi-class problem too. There are a lot of machine learning algorithms for classification in machine learning. Some algorithms are as follows:

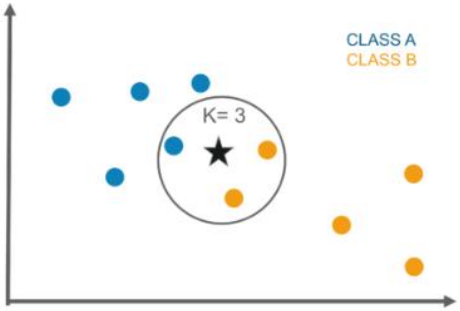
LOGISTIC REGRESSION:

It is a classification algorithm in machine learning that uses one or more independent variables to determine an outcome. The result is estimated with a dichotomous variable importance it will have just two potential results. The objective of logistic regression is to find a best-fitting relationship between the dependent variable and a set of independent variables.



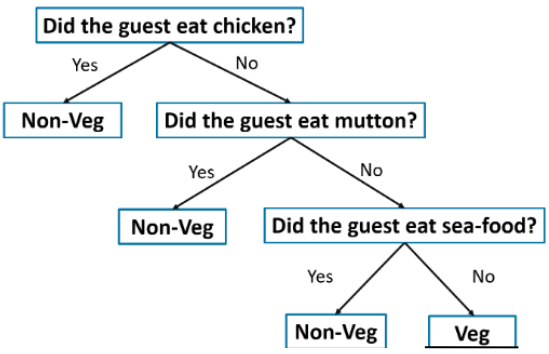
K-NEARNEST NEIGHBOUR:

It is a lazy learning algorithm that stores all instances corresponding to training data in n-dimensional space**.** It is a lazy learning algorithm as it does not focus on constructing a general internal model; instead, it works on storing instances of training data.



DECISION TREE:

The decision tree algorithm builds the classification model in the form of a tree structure. It utilizes the if-then rules which are equally exhaustive and mutually exclusive in classification. The process goes on with breaking down the data into smaller structures and eventually associating it with an incremental decision tree. The final structure looks like a tree with nodes and leaves. The rules are learned sequentially using the training data one at a time. Each time a rule is learned, the tuples covering the rules are removed. The process continues on the training set until the termination point is met.



(As the topic is based on Decision Tree Algorithm we will broadly describe it here)

DECISION TREE ALGORITHM:

DEFINITION:
A Decision Tree is a Supervised Machine Learning algorithm which looks like an inverted tree, wherein each node represents a predictor variable (feature), the link between the nodes represents a Decision and each leaf node represents an outcome (response variable).

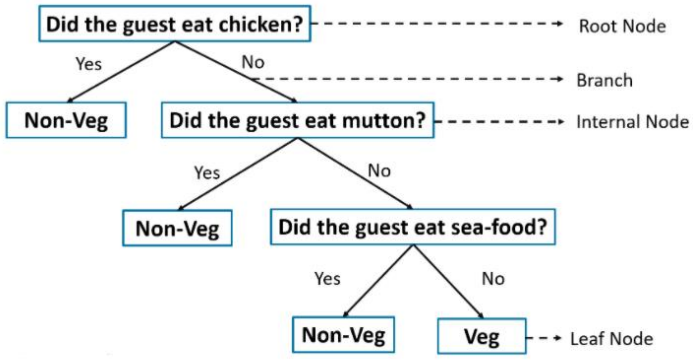
A Decision Tree has the following structure:

Root Node: The root node is the starting point of a tree. At this point, the first split is performed.

Internal Nodes: Each internal node represents a decision point (predictor variable) that eventually leads to the prediction of the outcome.

Leaf/ Terminal Nodes: Leaf nodes represent the final class of the outcome and therefore they're also called terminating nodes.

Branches: Branches are connections between nodes, they're represented as arrows. Each branch represents a response such as yes or no.



WORKING OF DECISION TREE ALGORITHM:

Step 1: Select the feature (predictor variable) that best classifies the data set into the desired classes and assign that feature to the root node.

Step 2: Traverse down from the root node, whilst making relevant decisions at each internal node such that each internal node best classifies the data.

Step 3: Route back to step 1 and repeat until you assign a class to the input data.

PRACTICAL IMPLEMENTION OF DECISION TREE ALGORITHM IN PYTHON:

PROBLEM STATEMENT:

To study a dataset of historic data of patients, and their response to different medications by using the trained decision tree to predict the class of an unknown patient, or to find a proper drug for a new patient.

DATASET DESCRIPTION:

The given dataset is about a set of patients, all of whom suffered from the same illness. During their course of treatment, each patient responded to one of 5 medications, Drug A, Drug B, Drug C, Drug X and Y.

Objective of the project is to build a model to find out which drug might be appropriate for a future patient with the same illness. The feature sets of this dataset are Age, Sex, Blood Pressure, and Cholesterol of patients, and the target is the drug that each patient responded to.

Age	Sex	BP	Cholesterol	Na_to_K	Drug
23	F	HIGH	HIGH	25.355	drugY
47	M	LOW	HIGH	13.093	drugC
47	M	LOW	HIGH	10.114	drugC
28	F	NORMAL	HIGH	7.798	drugX
61	F	LOW	HIGH	18.043	drugY
22	F	NORMAL	HIGH	8.607	drugX
49	F	NORMAL	HIGH	16.275	drugY
41	M	LOW	HIGH	11.037	drugC
60	M	NORMAL	HIGH	15.171	drugY
43	M	LOW	NORMAL	19.368	drugY
47	F	LOW	HIGH	11.767	drugC
34	F	HIGH	NORMAL	19.199	drugY
43	M	LOW	HIGH	15.376	drugY
74	F	LOW	HIGH	20.942	drugY
50	F	NORMAL	HIGH	12.703	drugX
16	F	HIGH	NORMAL	15.516	drugY
69	M	LOW	NORMAL	11.455	drugX
43	M	HIGH	HIGH	13.972	drugA
23	M	LOW	HIGH	7.298	drugC
32	F	HIGH	NORMAL	25.974	drugY
57	M	LOW	NORMAL	19.128	drugY
63	M	NORMAL	HIGH	25.917	drugY
47	M	LOW	NORMAL	30.568	drugY
48	F	LOW	HIGH	15.036	drugY
33	F	LOW	HIGH	33.486	drugY
28	F	HIGH	NORMAL	18.809	drugY
31	M	HIGH	HIGH	30.366	drugY

IMPLEMENTATION:

Now as the objective of the project is known let's get started.

STEP 1: IMPORT THE LIBRARIES:

```
import numpy as np
import pandas as pd
from sklearn.tree import DecisionTreeClassifier
```

- We have imported the Numpy library to convert our data into array form before fitting it to the algorithm to avoid shape issues.
- Pandas library lets us to manage, store and manipulate our dataset values.
- DecisionTreeClassifier is our main library as it is the algorithm we used to get the result in this following project work.

STEP 2: IMPORT THE DATASET:

```
[ ]  df=pd.read_csv("/content/drug200.csv")
```

- The dataset was downloaded from canvas as provided by our mentor Dr. Somdatta Chakrabortty ma'am. We used our system path to store the dataset and we read the dataset from there itself.

STEP 3: DATA PREPROCESSING:

```
from sklearn import preprocessing
le_sex = preprocessing.LabelEncoder()
le_sex.fit(['F','M'])
X[:,1] = le_sex.transform(X[:,1])


le_BP = preprocessing.LabelEncoder()
le_BP.fit([ 'LOW', 'NORMAL', 'HIGH'])
X[:,2] = le_BP.transform(X[:,2])


le_Chol = preprocessing.LabelEncoder()
le_Chol.fit([ 'NORMAL', 'HIGH'])
X[:,3] = le_Chol.transform(X[:,3])

X[0:5]
```

```
array([[23, 0, 0, 0, 25.355],
       [47, 1, 1, 0, 13.093],
       [47, 1, 1, 0, 10.113999999999999],
       [28, 0, 2, 0, 7.797999999999999],
       [61, 0, 1, 0, 18.043]], dtype=object)
```

STEP 4: DATA EXPLORATION AND ANALYSIS:

```
my_data = pd.read_csv("drug200.csv", delimiter=",")
my_data[0:5]
```

	Age	Sex	BP	Cholesterol	Na_to_K	Drug
0	23	F	HIGH	HIGH	25.355	drugY
1	47	M	LOW	HIGH	13.093	drugC
2	47	M	LOW	HIGH	10.114	drugC
3	28	F	NORMAL	HIGH	7.798	drugX
4	61	F	LOW	HIGH	18.043	drugY

```
[5] X = my_data[['Age', 'Sex', 'BP', 'Cholesterol', 'Na_to_K']].values
    X[0:5]
```

```
array([[23, 'F', 'HIGH', 'HIGH', 25.355],
       [47, 'M', 'LOW', 'HIGH', 13.093],
       [47, 'M', 'LOW', 'HIGH', 10.113999999999999],
       [28, 'F', 'NORMAL', 'HIGH', 7.797999999999999],
       [61, 'F', 'LOW', 'HIGH', 18.043]], dtype=object)
```

STEP 5: DATA TRAINING AND TESTING:

```
[8]  from sklearn.model_selection import train_test_split

     X_trainset, X_testset, y_trainset, y_testset = train_test_split(X, y, test_size=0.3, random_state=3)

[10] drugTree = DecisionTreeClassifier(criterion="entropy", max_depth = 4)
     drugTree # it shows the default parameters

     DecisionTreeClassifier(ccp_alpha=0.0, class_weight=None, criterion='entropy',
                            max_depth=4, max_features=None, max_leaf_nodes=None,
                            min_impurity_decrease=0.0, min_impurity_split=None,
                            min_samples_leaf=1, min_samples_split=2,
                            min_weight_fraction_leaf=0.0, presort='deprecated',
                            random_state=None, splitter='best')

[11] drugTree.fit(X_trainset,y_trainset)

     DecisionTreeClassifier(ccp_alpha=0.0, class_weight=None, criterion='entropy',
                            max_depth=4, max_features=None, max_leaf_nodes=None,
                            min_impurity_decrease=0.0, min_impurity_split=None,
                            min_samples_leaf=1, min_samples_split=2,
                            min_weight_fraction_leaf=0.0, presort='deprecated',
                            random_state=None, splitter='best')
```

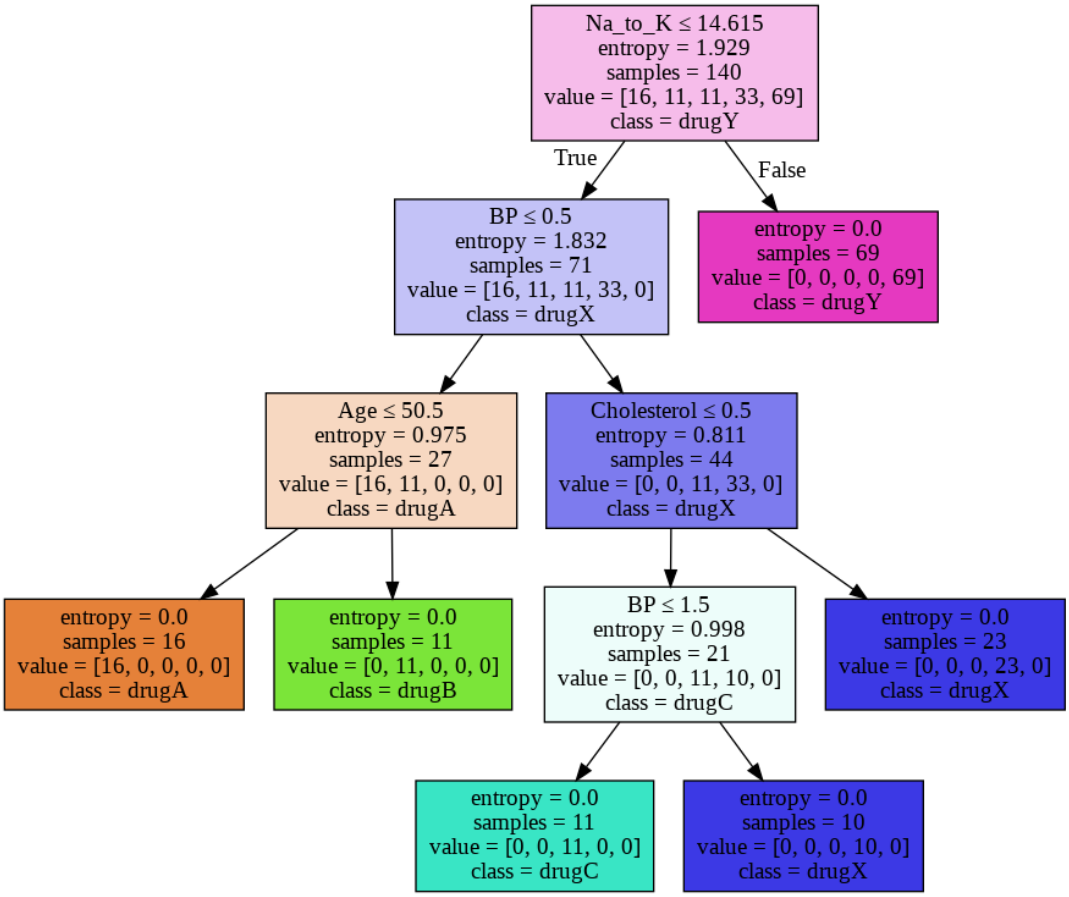
STEP 6: DATA BUILDING AND VISUALIZATION:

```
from sklearn.externals.six import StringIO
import pydotplus
import matplotlib.image as mpimg
from sklearn import tree
%matplotlib inline

/usr/local/lib/python3.7/dist-packages/sklearn/externals/six.py:31: FutureWarning: The module is deprecated in version 0.21 and will be removed in version 0.23 since we'v
  "(https://pypi.org/project/six/).", FutureWarning)

[17] dot_data = StringIO()
     filename = "drugtree.png"
     featureNames = my_data.columns[0:5]
     targetNames = my_data["Drug"].unique().tolist()
     out=tree.export_graphviz(drugTree,feature_names=featureNames, out_file=dot_data, class_names= np.unique(y_trainset), filled=True,  special_characters=True,rotate=False)
     graph = pydotplus.graph_from_dot_data(dot_data.getvalue())
     graph.write_png(filename)
     img = mpimg.imread(filename)
     plt.figure(figsize=(100, 200))
     plt.imshow(img,interpolation='nearest')
```

RESULT

```
[15] from sklearn import metrics
     import matplotlib.pyplot as plt
     print("DecisionTrees's Accuracy: ", metrics.accuracy_score(y_testset, predTree))
```

```
DecisionTrees's Accuracy:  0.9833333333333333
```



CONCLUSION

A decision tree is a predictive model which is a mapping from observations about an item to conclusions about its target value. Here in the above experiment, we achieve an accuracy of 98.33 after passing it through a range of values. The tree creates a visual representation of all possible outcomes, rewards and follow-up decisions in one document. Each subsequent decision resulting from the original choice is also depicted on the tree, so you can see the overall effect of any one decision. As you go through the tree and make choices, you will see a specific path from one node to another and the impact a decision made now could have down the road.

REFERENCES

- https://github.com
- https://kaggle.com
- www.javatpoint.com/decisiontree
- www.geeksforgeeks.org/machine-learning/decisiontree
- https://www.edureka.co/