Performance Evaluation of Decision Trees with Machine Learning Algorithm

Kiran Nitin Babar
AI&DS Department
Datta Meghe College of Engineering
Navi-Mumbai, India
kiran.babar@dmce.ac.in

Deepali Sanjay Chavan
AI&DS Department
Datta Meghe College of Engineering
Navi-Mumbai, India
deepali.chavan@dmce.ac.in

***Abstract*—**Decision tree learning is a supervised learning approach used in statistics, data mining and machine learning. Decision trees are considered to be one of the most popular approaches for representing classifiers. Researchers from various disciplines such as statistics, machine learning, pattern recognition and Data Mining have dealt with the issue of growing a decision tree from available data. Decision trees in machine learning will be used for classification problems, to categorize objects to gain an understanding of similar features. Decision trees helps in decision-making by representing complex choices in a hierarchical structure. Every node in decision tree verifies specific attributes, guiding decisions based on different data values in the dataset. Leaf nodes provide final outcomes and result which gives a clear and interpretable path for decision analysis in machine learning. Therefore implementation of Decision tree algorithm using python is presented in this paper

Keywords— Decision Trees, Classification. Machine learning, statistics, regression

1. INTRODUCTION

Data is often associated with uncertainty because of measurement inaccuracy, sampling discrepancy, outdated data sources, or other errors. In recent years, uncertain data has become ubiquitous because of new technologies for collecting data which can only measure and collect the data in an imprecise way. While many applications lead to data which contains errors, we refer to uncertain data sets as those in which the level of uncertainty can be quantified in some way. Many scientific measurement techniques are inherently imprecise. In such cases, the level of uncertainty may be derived from the errors in the underlying instrumentation. Many new hardware technologies such as sensors generate data which is imprecise. In such cases, the error in the sensor network readings can be modeled, and the resulting data can be modeled as imprecise data. In many applications such as the tracking of mobile objects, the future trajectory of the objects is modeled by forecasting techniques. Small errors in current readings can get magnified over the forecast into the distant future of the trajectory. This is frequently encountered in cosmological applications when one models the probability of encounters with Near-Earth-Objects (NEOs). Errors in forecasting are also encountered in non-spatial applications such as electronic commerce. In many applications such as privacy- preserving data mining, the data is modified by adding perturbations to it. In such cases, the format of the output is exactly the same as that of uncertain data. Location- based services: in the scenario of moving objects (such as vehicles or people), it is impossible for the database to track the exact locations of all objects at all time instants. Therefore, the location of each object is associated with uncertainty between updates. In recent years, there has been much research on the management of uncertain data in databases, such as the representation of uncertainty in databases and querying data with uncertainty. However, little research work has addressed the issue of mining uncertain data. We know that with uncertainty, data values are no longer atomic. To apply traditional data mining techniques, uncertain data has to be summarized into atomic values.

2. RELATED WORKS

There has been a growing interest in uncertain data mining and recently more research has been conducted on that. Most of them focus on clustering uncertain data [1], [2]. The key idea is that when computing the distance between two uncertain objects, the probability distributions of objects are used to compute the expected distance. In [3], the well-known k-means clustering algorithm is extended to the UK-means algorithm for clustering uncertain data. Data uncertainty is usually captured by pdf's, which are generally represented by sets of sample values. Mining uncertain data is therefore computationally costly due to information explosion for sets of samples vs. single values. To improve the performance of UK-means, pruning techniques have been proposed [4]. Xia et al. [5] introduce a new conceptual clustering algorithm for uncertain categorical data. Agarwal [6] proposes density based transforms for uncertain data mining. There is also some research on identifying frequent item sets and association mining from uncertain datasets [7]. The support of item sets and confidence of association rules are integrated with the existential probability of transactions and items. Burdicks

[8] discuss OLAP computation on uncertain data. None of them address the issue of developing a general classification and prediction algorithm for uncertain data. Decision trees are one of the most important aspects for "Decision-making". Classification is one of the most widespread data mining problems found in real life. Decision tree classification is one of the best-known solution approaches [9], [10], [11]. In C4.5 and probabilistic decision trees, missing values in training data are handled by using fractional tuples [12]. During testing, each missing value is replaced by multiple values with probabilities based on the training tuples, thus allowing probabilistic classification results. They have adopted the technique of fractional tuple for splitting tuples into subsets when the domain of its pdf spans across the split point.

3. EXISTING SYSTEM

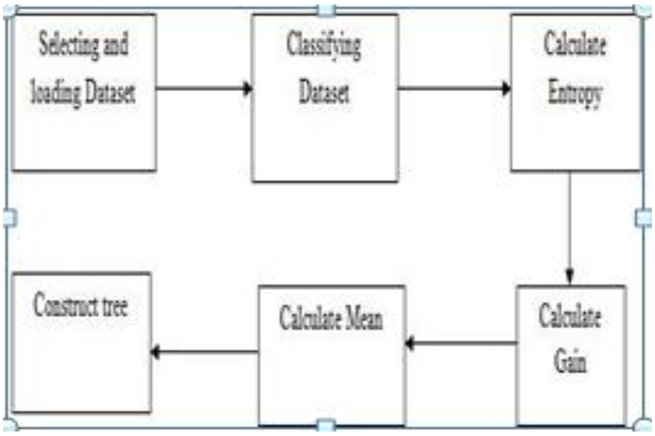
In traditional decision-tree classification, a feature or an attribute of a tuple is either categorical or numerical. For the latter, a precise and definite point value is usually assumed. In many applications, however, data uncertainty is common. The value of a feature/attribute is thus best captured not by a single point value, but by a range of values giving rise to a probability distribution. Although the previous techniques can improve the efficiency of means, they do not consider the spatial relationship among cluster representatives, nor make use of the proximity between groups of uncertain objects to perform pruning in batch. A simple way to handle data uncertainty is to abstract probability distributions by summary statistics such as means and variances called averaging approach. Another approach is to consider the complete information carried by the probability distributions to build a decision tree called Distribution-based approach.

4. PROPOSED SYSTEM

The main goal of the decision tree is to construct by using:

(1) The basic algorithm to construct decision trees out of uncertain datasets.
(2) Find out whether the Distribution-based approach could lead to higher classification accuracy compared with the Averaging approach.
(3) Implementation of Decision tree using python showing types of tumour for breast cancer dataset.

The following figure 2 shows the flow diagram of the proposed methodology.



1: Proposed Methodology

The two approaches for handling uncertain data. The first approach, called "Averaging", transforms an uncertain dataset to a point-valued one by replacing each pdf with its mean value. To exploit the full information carried by the pdf's, the second approach, called "Distribution- based", considers all the sample points that constitute each pdf.

A. AVERAGING

A simple way to handle data uncertainty is to abstract probability distributions by summary statistics such as means and variances which is called as the averaging approach. A straightforward way to deal with the uncertain information is to replace each pdf with its expected value, thus effectively converting the data tuples to point-valued tuples. The algorithm starts with the root node and with S being the set of all training tuples. At each node n, we first check if all the tuples in S have the same class label.

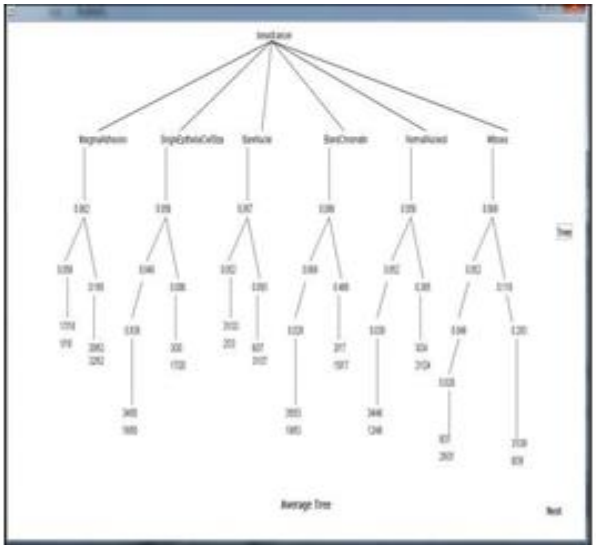


Figure 2: Averaging tree

B. DISTRIBUTION BASED

An approach is to consider the complete information carried by the probability distributions to build a decision tree which is called as Distribution-based approach. After an attribute Ajn and a split point zn has been chosen for a node n, we split the set of tuples S into two subsets L and R. The major difference from the point-data case lies in the way the set S is split. If the pdf properly contains the split point, i.e., ai,jn ≤ zn < bi,jn, we split ti into two fractional tuples[3] tL and tR and add them to L and R, respectively. The algorithm is called UDT (Uncertain Decision Tree).

5. DECISION TREE ALGORITHMS

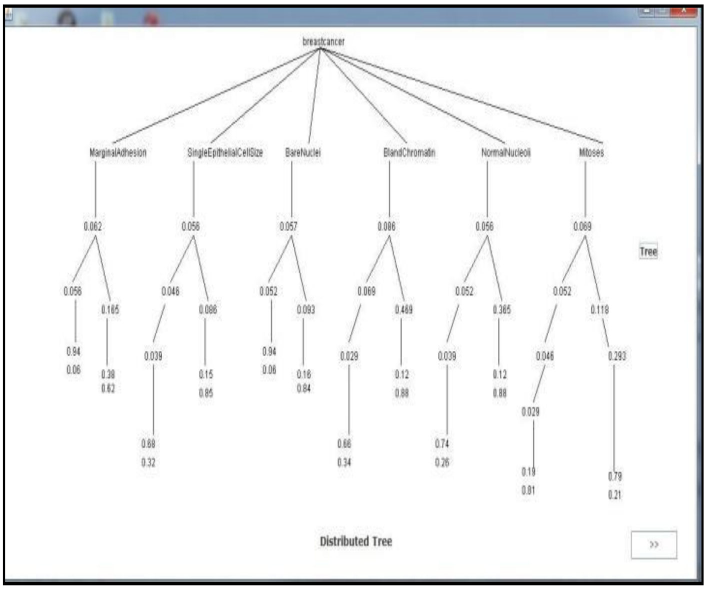
Decision trees can run varied algorithms to divide and subdivide a node into further sub-nodes. The decision tree uses all the available variables to split the nodes but eventually chooses the split that yields the most homogeneous sub-nodes.

1. Iterative dichotomiser 3 (ID3)

The Iterative dichotomiser 3 algorithm generates decision trees with the whole dataset ‘X’ as the root node. It then repeats the instructions on each attribute and uses metrics like entropy or information gain to divide the information into subsets. Upon splitting, the algorithm recurses on every subset by considering the attributes not considered before in the iterated ones.

2. C4.5

C4.5 is an advanced version of the ID3 algorithm. It considers classified samples as data. The algorithm uses normalized information gain to carry out the splitting of the nodes. The feature having the highest information gain makes the final decision on the data split.C4.5 manages both discrete and continuous attributes efficiently.



3. Classification and regression trees (CART)

The CART algorithm solves both regression and classification problems. Also, it creates decision points by using the Gini index metric, unlike the ID3 and C4.5 algorithms that use information gain or entropy and gain ratio for splitting the datasets.

4. Chi-square automatic interaction detector (CHAID)

The CHAID algorithm reveals the relationship between variables of all types, including nominal, ordinal, or continuous. The CHAID approach creates a tree that identifies how variables can best merge to disclose the outcome for the given dependent variable.

6.RESULTS AND DISCUSSION

The accuracy of a decision tree classifier can be much improved if the “complete information” of a data item (taking into account the probability density function (pdf)) is utilised. Distribution based algorithm can improve classification accuracy because there are more choices of split points. The distribution approach has to examine k (ms-1) split points whereas the AVG approach has to examine k(m-1) split points. Entropy calculations are the most computation intensive part of UDT. To explore the potential of achieving a higher classification accuracy by considering data uncertainty, we have implemented AVG and UDT and applied them to 04 datasets namely glass dataset, page block , Japanese Vowel, Breast Cancer dataset taken from the UCI Machine Learning Repository. These datasets are chosen because they contain mostly numerical attributes obtained from measurements. We model uncertainty information by fitting appropriate error models on to the point data. For each tuple ti and for each attribute Aj , the point value vi;j reported in a dataset is used as the mean of a pdf fi;j , defined over an interval [ai;j ; bi;j]. The range of values for Aj (over the whole data set) is noted and the width of [ai;j ; bi;j] is set to w _ jAj j, where jAj j denotes the width of the range for Aj and w is a controlled parameter. To generate the pdf fi;j , we consider two options. The first is uniform distribution, which implies fi;j(x) = (bi;j - ai;j)-1. The other option is Gaussian distribution, for which we use 1/4 (bi;j -ai;j) as the standard deviation. In both cases, the pdf is generated using s sample points in the interval. Using this method (with controllable parameters w and s, and a choice of Gaussian vs. uniform distribution), we transform a data set with point values into one with uncertainty. The reason that we choose

Gaussian distribution and uniform distribution is that most physical measures involve random noise which follows Gaussian distribution, and that digitisation of the measured values introduces quantisation noise that is best described by a uniform distribution.

TABLE 1:

Accuracy Improvement by Considering the Distribution

DATASET	AVG	BEST CASE	UDT (Gaussian Distribution)
Japanese Vowel	81.89	87.30	87.30
Page Block	95.73	96.82	96.82
Glass	66.49	72.75	69.60
Breast Cancer	93.52	95.93	94.73

From the table, we see that UDT builds more accurate decision trees than AVG does for different distributions over a wide range of w. For the first data set, whose pdf is modelled from the raw data samples, the accuracy is improved from 81.89% to 87.30%. Also the following accuracy classification graph shows UDT gives better accuracy than averaging.

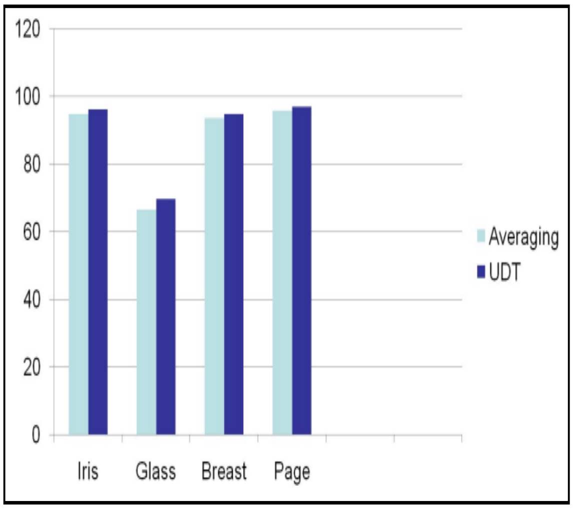
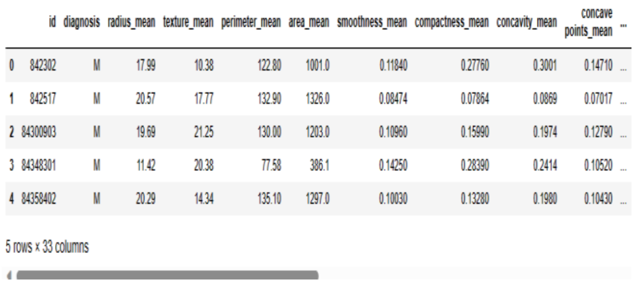


Figure 4: Graph of Accuracy classification

7. IMPLEMENTATION & RESULTS

Python Implementation of Decision Tree Algorithm:

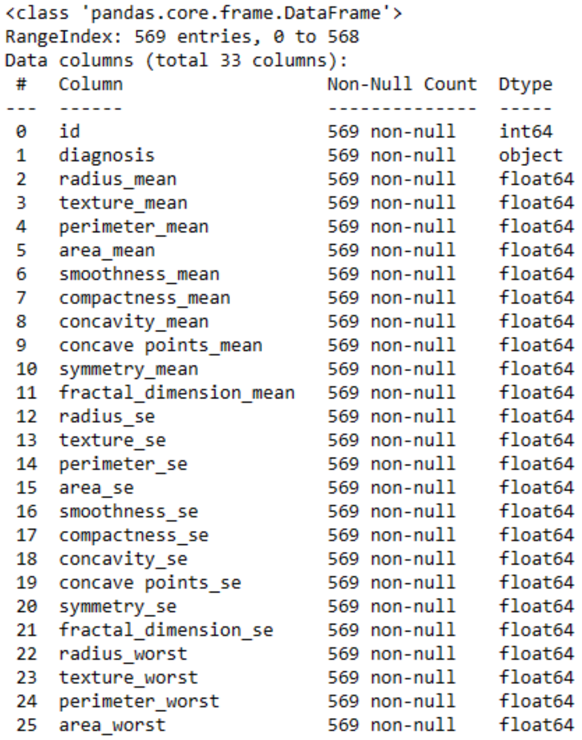
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt


import os
for dirname, _, filenames in
os.walk('C:/Users/DMCE/Desktop/data.csv'):
    for filename in filenames:
        print(os.path.join(dirname, filename))

dataset =
pd.read_csv("C:/Users/DMCE/Desktop/Kiran/data.csv")

dataset.head()
```



	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	concavity_mean	concave points_mean	...
0	842302	M	17.99	10.38	122.80	1001.0	0.11840	0.27760	0.3001	0.14710	...
1	842517	M	20.57	17.77	132.90	1326.0	0.08474	0.07864	0.0869	0.07017	...
2	84300903	M	19.69	21.25	130.00	1203.0	0.10960	0.15990	0.1974	0.12790	...
3	84348301	M	11.42	20.38	77.58	386.1	0.14250	0.28390	0.2414	0.10520	...
4	84358402	M	20.29	14.34	135.10	1297.0	0.10030	0.13280	0.1980	0.10430	...

5 rows × 33 columns

dataset.info()

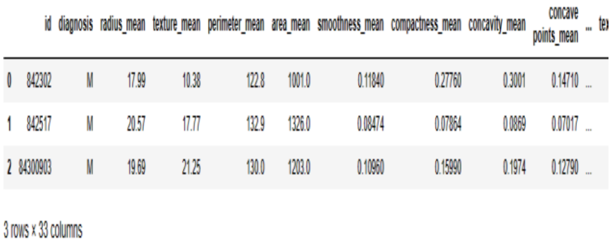


```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 569 entries, 0 to 568
Data columns (total 33 columns):
 #   Column                   Non-Null Count  Dtype  
---  ------                   --------------  -----  
 0   id                       569 non-null    int64  
 1   diagnosis                569 non-null    object 
 2   radius_mean              569 non-null    float64
 3   texture_mean             569 non-null    float64
 4   perimeter_mean           569 non-null    float64
 5   area_mean                569 non-null    float64
 6   smoothness_mean          569 non-null    float64
 7   compactness_mean         569 non-null    float64
 8   concavity_mean           569 non-null    float64
 9   concave points_mean      569 non-null    float64
 10  symmetry_mean            569 non-null    float64
 11  fractal_dimension_mean   569 non-null    float64
 12  radius_se                569 non-null    float64
 13  texture_se               569 non-null    float64
 14  perimeter_se             569 non-null    float64
 15  area_se                  569 non-null    float64
 16  smoothness_se            569 non-null    float64
 17  compactness_se           569 non-null    float64
 18  concavity_se             569 non-null    float64
 19  concave points_se        569 non-null    float64
 20  symmetry_se              569 non-null    float64
 21  fractal_dimension_se     569 non-null    float64
 22  radius_worst             569 non-null    float64
 23  texture_worst            569 non-null    float64
 24  perimeter_worst          569 non-null    float64
 25  area_worst               569 non-null    float64
```

dataset = dataset.drop(["id"], axis = 1)

dataset = dataset.drop(["Unnamed: 32"], axis = 1)

dataset.head(3)



	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	concavity_mean	concave points_mean	...	te
0	842302	M	17.99	10.38	122.8	1001.0	0.11840	0.27760	0.3001	0.14710	...	
1	842517	M	20.57	17.77	132.9	1326.0	0.08474	0.07864	0.0869	0.07017	...	
2	84300903	M	19.69	21.25	130.0	1203.0	0.10960	0.15990	0.1974	0.12790	...	

3 rows × 33 columns

plt.title("Malignant vs Benign Tumor")

plt.xlabel("Radius Mean")

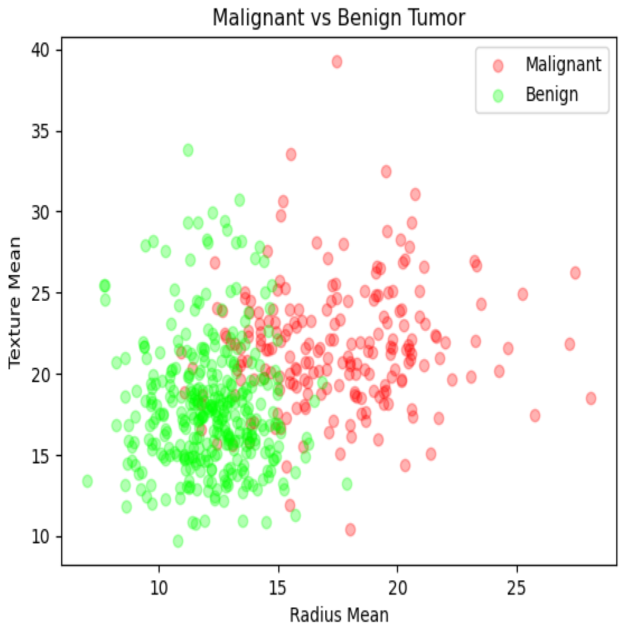
plt.ylabel("Texture Mean")

plt.scatter(M.radius_mean, M.texture_mean, color = "red", label = "Malignant", alpha = 0.3)

plt.scatter(B.radius_mean, B.texture_mean, color = "lime", label = "Benign", alpha = 0.3)

plt.legend()

plt.show()



dataset.diagnosis = [1 if i == "M" else 0 for i in dataset.diagnosis]

x = dataset.drop(["diagnosis"], axis = 1)

y = dataset.diagnosis.values

x = (x - np.min(x)) / (np.max(x) - np.min(x))

from sklearn.model_selection import train_test_split

x_train, x_test, y_train, y_test = train_test_split(x, y, test_size = 0.3, random_state = 42)

from sklearn.tree import DecisionTreeClassifier

dt = DecisionTreeClassifier()

```
dt.fit(x_train, y_train)

dt.score(x_test, y_test)
```

Accuracy: 0.8538011695906432

8.CONCLUSION

The model of decision-tree classification using breast cancer dataset has been extended to accommodate data tuples having numerical attributes with uncertainty described by arbitrary pdf's. We have modified classical decision tree building algorithms to build decision trees for classifying such data. It has been found empirically that when suitable pdf's are used, exploiting data uncertainty leads to decision trees with remarkably higher accuracies. Performance is an issue, though, because of the increased amount of information to be processed, as well as the more complicated entropy computations involved. Although our novel techniques are primarily designed to handle uncertain data, they are also useful for building decision trees using classical algorithms when there are tremendous amounts of data tuples.

9.REFERENCES

[1] Michael Chau, Reynold Cheng, and Ben Kao "Uncertain Data Mining: A New Research Direction", published in Proceedings of the 10th Pacific-Asia conference on Advances in Knowledge discovery and Data Mining.

[2] Smith Tsangy ,Ben Kao, Kevin Y. Yip, Wai-Shing Ho, Sau Dan Lee "Decision Trees for Uncertain Data" Knowledge and Data Engineering, IEEE Transactions on Volume 23, Issue 1,Jan 2011 pp.64-78

[3] Varsha Choudhary, Pranita Jain, "Classification: A Decision Tree for Uncertain Data Using CDF" International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622 Vol. 3, Issue 1, January -February 2013, pp.1501-1506

[4] W. Street, W. Wolberg, and O. Mangasarian, "Nuclear feature extraction for breast tumor diagnosis," in SPIE, vol. 1905, San Jose, CA, U.S.A., 1993, pp. 861–870. [Online]. Available: http://citeseer.ist.psu.edu/street93nuclear.html

[5] M. Chau, R. Cheng, B. Kao, and J. Ng, "Uncertain data mining: An example in clustering location data," in PAKDD, ser. Lecture Notes in Computer Science, vol. 3918. Singapore: Springer, 9–12 Apr. 2006, pp.199–204.

[6] J.R, QUINLAN 'Induction of Decision Trees" Machine Learning 1: 81-106.

[7] R.Agrawal, T.Imielinski and A.N Swami. Database Mining: A performance perspective. IEEE Trans.
Knowl. Data Eng., vol.5, no.6, pp. 914-925, 1993

[8] Y. Sahin and E. Duman "Detecting Credit Card Fraud by Decision Trees and Support Vector Machines" Proceedings of the International MultiConference of Engineers and Computer Scientists 2011 Vol-I IMECS 2011.

[9] Hawarah L, Simonet A, Simonet M "Dealing with Missing Values in a Probabilistic Decision Tree during Classification", The Second Intern8ational Workshop on Mining Complex Data, pp. 325-329.

[10] Data Mining Analysis (breast-cancer data) Jung-Ying Wang Register number: D9115007, May, 2003.

[11] Khumesh Patil, Namrata Pagare, Pallavi Narkhede, Prashant Brahmankar " Classifying Climate Data (uncertain) using Decision Tree" International Journal of Advanced Research (2014), Volume 2, Issue 4, 402-408 ,ISSN 2320-5407.

[12] Nikita Patel , Saurabh Upadhyay" Study of Various Decision Tree Pruning Methods with their Empirical Comparison in WEKA" International Journal of Computer Applications (0975 – 8887) Volume 60–
No.12, December 2012

[13]https://www.kaggle.com/datasets/uciml/breast-cancer-Wisconsin-data