

PROFESSIONAL TRAINING REPORT
at
Sathyabama Institute of Science and Technology
(Deemed to be University)

Submitted in partial fulfillment of the requirements for the award of Bachelor of
Technology Degree in Information Technology

By
SHAURYA SAMEER
Registration No.: 39120110



DEPARTMENT OF INFORMATION TECHNOLOGY
SCHOOL OF COMPUTING
SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY

JEPPIAR NAGAR, RAJIV GANDHI SALAI,
CHENNAI-600119, TAMIL NADU

NOVEMBER 2021



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)
Accredited with Grade "A" by NAAC

(Established under Section 3 of UGC Act, 1956)

JEPPIAAR NAGAR, RAJIV GANDHI SALAI

CHENNAI- 600119

www.sathyabama.ac.in



DEPARTMENT OF INFORMATION TECHNOLOGY

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **SHAURYA SAMEER (Reg. No: 39120110)** who carried out the project entitled "**Divorce Prediction using Decision Tree (Machine Learning)**" under my supervision from June 2021 to November 2021.

Internal Guide

Dr. S. Revathy

Head of the Department

Dr. R. Subhashini

Submitted for Viva Voce Examination held on _____

Internal Examiner

External Examiner

DECLARATION

I, **SHAURYA SAMEER** hereby declare that the project entitled **Divorce Prediction Using Decision Trees (Machine Learning)** completed by me under the guidance of **Dr. S. Revathy** is submitted in partial fulfilment of the requirements for the award of Bachelor in Technology Degree in Information Technology.

DATE:

SHAURYA SAMEER

PLACE:

DIGITAL SIGNATURE OF CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management** at **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. Sasikala M.E., Ph.D, Dean**, School of Computing, **Dr. R. Subhashini, M.E., Ph.D. Head of the Department of Information Technology** for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr. S. Revathy, M.E., Ph.D.**, for her valuable guidance, suggestions and constant encouragement that paved the way for the successful completion of my project work.

I wish to express my thanks to all teaching and non-teaching staff members of the **Department of Information Technology** who were helpful in many ways throughout the completion of the project.

TRAINING CERTIFICATE



ABSTRACT

Decision Tree Analysis is a predictive modelling tool that has applications spanning a number of different areas. In general, decision trees are constructed via an algorithmic approach that identifies ways to split a dataset based on different conditions. Decision Trees are a non-parametric supervised learning method used for both classification and regression tasks. It is one of the most widely used and practical methods for supervised learning. The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features.

Our particular dataset contains data of about 150 couples with their corresponding Divorce Predictors Scale variables (DPS) on the basis of Gottman couples therapy. The couples are from various regions of Turkey wherein the records were acquired from face-to-face interviews from couples who were already divorced or happily married. All responses were collected on a 5 point scale (0=Never, 1=Seldom, 2=Averagely, 3=Frequently, 4=Always). The input features are various life decisions and situations with categorical values depicting whether the particular problem statement is true for a particular couple or not.

This project aims to address a particular dataset based on divorce outcomes in different cases (life choices, compatibility etc.) and train a machine learning model using decision trees to provide predictions on whether a couple will get divorced or not based on certain some input features. Decision tree offers very high precision compared to other machine learning algorithms while being comparatively easier to use and they shine the most in categorical datasets like our particular case and our dataset is a perfect fit for it.

TABLE OF CONTENTS		
CHAPTER NO.	TITLE	PAGE NO.
	ABSTRACT	i
	LIST OF FIGURES	ii
1	INTRODUCTION	1
1.1	ABOUT THE CONCEPT	1
1.2	IMPORTANT TERMINOLOGY RELATED TO DECISION TREES	2
2	AIM AND SCOPE OF THE PRESENT INVESTIGATION	3
2.1	AIM	5
2.2	SCOPE	5
2.2.1	ACCESSIBILITY	6
3	EXPERIMENTAL OR MATERIALS AND METHODS, ALGORITHMS USED	7
3.1	HARDWARE/SOFTWARE REQUIREMENTS	7
3.1.1	WINDOWS/MAC/LINUX SYSTEM	7
3.1.2	ANACONDA NAVIGATOR-JUPYTER NOTEBOOK	7
3.2	DEFINING THE DATASET	7
3.3	ALGORITHM USED	10

3.3.1	LOADING THE DATASET	10
3.3.2	EXTRACTING THE TARGET VARIABLE FROM THE DATASET	11
3.3.3	SPLITTING THE DATA INTO TRAINING AND TESTING SETS	11
3.3.4	FITTING THE MODEL IN DECISION TREE CLASSIFIER	11
3.3.5	BUILDING AND SAVING OUR TREE AS A PNG FILE	12
4	PERFORMANCE METRICS AND RESULTS	14
4.1	PERFORMANCE METRICS	14
4.1.1	ACCURACY SCORE	14
4.1.2	CONFUSION MATRIX	15
4.2	RESULTS	16
5	SUMMARY AND CONCLUSION	17
5.1	FUTURE WORK	17
	REFERENCES	18
	APPENDIX	19

LIST OF FIGURES		
S. NO.	TITLE	PAGE NO.
1.1	Decision Tree Terminology	2
3.1	Loading Dataset	10
3.2	Desired Dataset	10
3.3	Extracting class attribute into a separate variable and dropping it from the table	11
3.4	Using train_test_split to split train and test datasets	11
3.5	Fitting our model	12
3.6	Plotting and saving the tree	13
3.7	Decision Tree	13
4.1	Checking the accuracy	14
4.2	Accuracy of our model	14
4.3	Confusion Matrix code	15
4.4	Confusion Matrix of our model	15
4.5	Score	16
	APPLICATION SNAPSHOTS	19
	APPLICATION SNAPSHOTS	20
	APPLICATION SNAPSHOTS	21

CHAPTER 1

INTRODUCTION

1.1 ABOUT THE CONCEPT

Classification is a two-step process, learning step and prediction step, in machine learning. In the learning step, the model is developed based on given training data. In the prediction step, the model is used to predict the response for given data. Decision Tree is one of the easiest and popular classification algorithms to understand and interpret.

Decision Tree algorithm belongs to the family of supervised learning algorithms. Unlike other supervised learning algorithms, the decision tree algorithm can be used for solving regression and classification problems too.

The goal of using a Decision Tree is to create a training model that can be used to predict the class or value of the target variable by learning simple decision rules inferred from prior data (training data).

In Decision Trees, for predicting a class label for a record we start from the root of the tree. We compare the values of the root attribute with the record's attribute. On the basis of comparison, we follow the branch corresponding to that value and jump to the next node.

Types of Decision Trees:

Types of decision trees are based on the type of target variable we have. It can be of two types:

1. **Categorical Variable Decision Tree:** Decision Tree which has a categorical target variable then it called a Categorical variable decision tree.
2. **Continuous Variable Decision Tree:** Decision Tree has a continuous target variable then it is called Continuous Variable Decision Tree.

1.2 IMPORTANT TERMINOLOGY RELATED TO DECISION TREES

- **Root Node:** It represents the entire population or sample and this further gets divided into two or more homogeneous sets.
- **Splitting:** It is a process of dividing a node into two or more sub-nodes.
- **Decision Node:** When a sub-node splits into further sub-nodes, then it is called the decision node.
- **Leaf / Terminal Node:** Nodes do not split is called Leaf or Terminal node.
- **Pruning:** When we remove sub-nodes of a decision node, this process is called pruning. You can say the opposite process of splitting.
- **Branch / Sub-Tree:** A subsection of the entire tree is called branch or sub-tree.
- **Parent and Child Node:** A node, which is divided into sub-nodes is called a parent node of sub-nodes whereas sub-nodes are the child of a parent node.

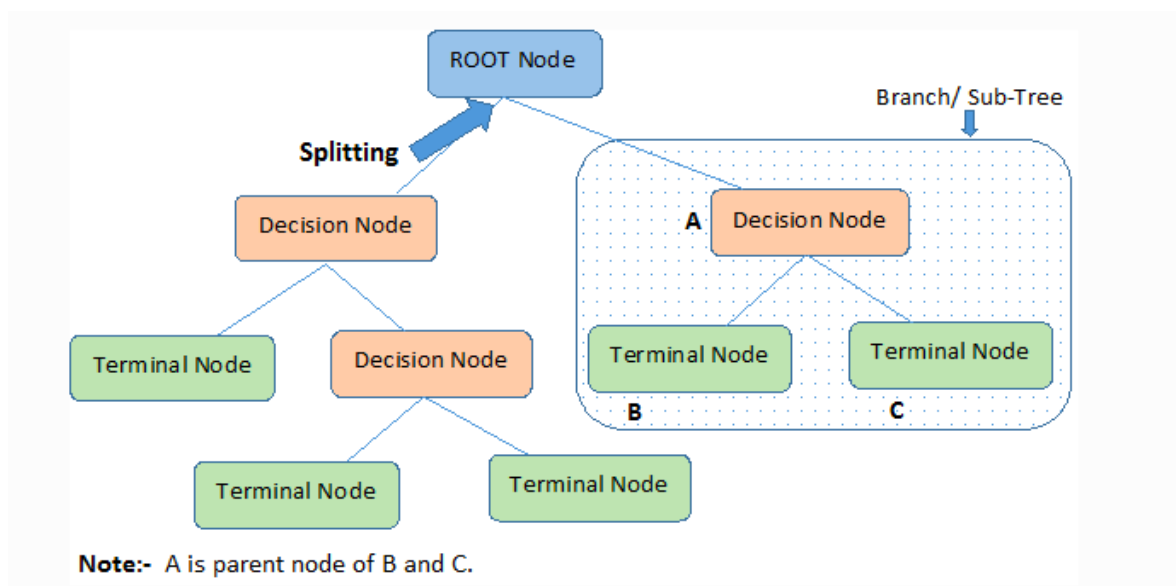


Fig: 1.1: Decision Tree Terminology

Decision trees classify the examples by sorting them down the tree from the root to some leaf/terminal node, with the leaf/terminal node providing the classification of the example. Each node in the tree acts as a test case for some attribute, and each edge descending from the node corresponds to the possible answers to the test case. This process is recursive in nature and is repeated for every sub tree rooted at the new node.

CHAPTER-2

AIM AND SCOPE OF THE PRESENT INVESTIGATION

The numbers of divorce cases are increasing very rapidly all over the world. In the last few decades, the numbers of divorces have gone up from 1 in 1000 to 13 in 1000 in India. Due to this reason, it is a major concern for marriage counsellors and therapists. Therefore, an effective divorce prediction technique is needed that helps a marriage counsellor or a therapist to identify how severe a case is. In this study, I present a study on divorce case prediction using an existing machine learning algorithm. I have applied the Decision Tree classifier for prediction of divorce cases, and determined the accuracy. The criterion employed in this study makes use of Gottman method to make the predictions. The algorithms after the training will predict whether the divorce will occur or not. This can help the therapist to analyse how tense the situation is between a couple and hence counsel them accordingly. We have achieved the highest accuracy of 94% with the Decision Tree model. The study can be expanded to a real-time marriage legal consultation automatically; a marriage legal dialogue system based on decision tree was designed in this study. Firstly, the legal consultation problem is transformed into a classification task. Secondly, a legal consultation classification prediction model based on the decision tree algorithm was trained by collected data. Finally, the model was tested and performance was noted. The proposed system works well in some real applications, such as the divorce problem, and the experimental results show that it is outperforming other algorithms and are more applicable than the other algorithms for this use case.

1) Decision Tree induction: Decision Tree induction is a classification pattern recognition approach that is applied in both CART and C5.0 algorithms. A decision tree is an inductive inference method widely used in a supervised classification learning technique for “[classifying] instances by sorting them down the tree from the root to some leaf node, which provides the classification of the instance. Each node in the tree specifies a test of some attribute of the instance, and each branch

descending from that node corresponds to one of the possible values for this attribute. An instance is classified by starting at the root node of the tree, testing the attribute specified by this node, then moving down the tree branch corresponding to the value of the attribute. This process is then repeated for the subtree rooted at the new node". The construction of a decision tree required two conceptual phases: growing and pruning. By its nature, the decision tree classifier is a greedy algorithm because the tree is grown by "recursively splitting the training set based on local optimal criteria until all or most of the records belonging to each of the partitions bearing the same class label". With that respect, the tree usually contains overfitting data. Overfitting the data occurs when a predictive model would classify perfectly the known data, but fail to classify anything useful on the yet-unseen data. Thus the attempt to make a tree not too closely taking on inaccurate data that can infect itself with substantial errors and reduce its predictive power is called pruning. The pruning phase handles the problem of over-fitting the data by removing the noise and outliers, which eventually increases the accuracy of the classification.

2) CART Algorithm CART, an abbreviation of Classification And Regression Trees, was first introduced by Breiman which is a binary tree using GINI Index as its splitting criteria. CART can handle both nominal and continuous attributes to construct a decision tree. Also, it can handle missing values by surrogating tests to approximate outcomes. In the pruning phase, CART uses prepruning technique called Cost – Complexity pruning to remove redundant branches from the decision tree to improve the accuracy. In Breiman's explanation, the Cost-Complexity pruning "proceeds in two stages. In the first stage, a sequence of increasingly smaller trees is built on the training data. In the second stage, one of these trees is chosen as the pruned tree, based on its classification accuracy on a pruning set. Pruning set is a portion of the training data that is set aside exclusively for pruning alone". In other words, CART adopts a cross –validated method in its pruning technique.

3) C5.0 Algorithm C5.0 algorithm is a successor of C4.5 algorithm. A few changes in C5.0 have made it to outperform C4.5 on the efficiency and the memory. Unlike

CART only gives a binary tree, C5.0 can generate multi-branch tree in a situation where one or more nominal inputs are given. Similar to its previous versions (C4.5 & ID3), C5.0 uses an information-based criterion (a.k.a Entropy or Information Gain) as an attribute selection measure to build a decision tree. For overfitting avoidance, C5.0 uses a pessimistic pruning approach, Rule-post pruning, to remove unreliable branches from the decision tree to reduce the size of the tree without any loss of its predictive accuracy. The Rule – post pruning starts off by converting a decision tree to an equivalent set of rules, then based on statistical confidence estimations for error rate it evaluates the rules with the aim of simplifying them without affecting the accuracy. Unlike CART pruning technique in which it uses a portion of the training set for validation, C5.0 pruning technique adopts the Binomial Confidence Limit method in which it allows all of the available labelled data to be used for training. In a case of handling missing values, C5.0 allows to whether estimate missing values as a function of other attributes or apportions the case probabilistically among the results.

2.1 AIM

Decision trees are able to generate understandable rules, perform classification without requiring much computation while being able to handle categorical variables and also provide clear indication of the hierarchy of fields in terms of importance. Our aim is to train a decision tree model for a Divorce Prediction dataset and evaluate it.

2.2 SCOPE

Divorce can bring several types of emotions to the forefront for a family, and the children involved are no different. Feelings of loss, anger, confusion, anxiety, and many others, all may come from this transition. Divorce can leave children feeling overwhelmed and emotionally sensitive. Compatibility in marriage is very important and our dataset helps a lot in determining that. With better preparation and compatibility prediction, couples can make informed decisions about their future

weighing in their differences and similarities to better know how well they'd fare together. With the help of machine learning and decision trees in particular, we can easily determine the factors that were the most important in several real life divorced relationship and use its results into making better future decisions.

Decision Trees are similar to the human decision-making process and so, it's pretty easy to understand. It can shine in both situations where the data input is either discrete or continuous. When data does not offer benefits while splitting, it directly stops the execution. Being transparent in nature is one of the best advantages of Decision Tree. Another main advantage is the ability to selecting the most biased feature and comprehensibility nature. It is also easy to classify and is Interpretable easily and is also used for both continuous and discrete data sets. Variable screening and feature selection are good enough in decision tree. Talking about its performance, non-linear does not affect any of the parameters of the decision tree.

2.2.1 Accessibility

There are already several software suites designed for ease of access and development of decision tree-based machine learning algorithms. Jupyter Notebook is well optimised for the use in data science related projects like these and Kaggle is one of the largest repositories available for machine learning. Decision tree algorithms are readily available and easy to setup and learn at this current date.

Decision tree is also very accessible to people who are not familiar with Machine Learning algorithms. Decision trees do not require feature scaling to be performed as they are not sensitive to the variance in the data. They are resistant to outliers by nature and do not require a lot of pre-processing in advance making them very accessible to use.

CHAPTER 3

METHODS OR ALGORITHMS USED

3.1 HARDWARE/SOFTWARE REQUIREMENT

- Windows/Mac/Linux Based System
- Anaconda Navigator – Jupyter Notebook
- Necessary Python Libraries (Pandas, Numpy, Sklearn, Graphviz etc)

3.1.1 *Windows/Mac/Linux System*

Windows, Mac or Linux based operating system for personal computers or any other internet of things devices.

3.1.2 *Anaconda Navigator – Jupyter Notebook*

Jupyter Notebook with any of the supported web browsers. It aims to support the latest versions of these browsers:

- Chrome
- Safari
- Firefox

Up to date versions of Opera and Edge may also work.

3.2 DEFINING THE DATASET

In this dataset, we have 170 samples with 54 input features, which are:

1. If one of us apologizes when our discussion deteriorates, the discussion ends.
2. I know we can ignore our differences, even if things get hard sometimes.
3. When we need it, we can take our discussions with my spouse from the

beginning and correct it.

4. When I discuss with my spouse, to contact him will eventually work.
5. The time I spent with my wife is special for us.
6. We don't have time at home as partners.
7. We are like two strangers who share the same environment at home rather than family.
8. I enjoy our holidays with my wife.
9. I enjoy traveling with my wife.
10. Most of our goals are common to my spouse.
11. I think that one day in the future, when I look back, I see that my spouse and I have been in harmony with each other.
12. My spouse and I have similar values in terms of personal freedom.
13. My spouse and I have similar sense of entertainment.
14. Most of our goals for people (children, friends, etc.) are the same.
15. Our dreams with my spouse are similar and harmonious.
16. We're compatible with my spouse about what love should be.
17. We share the same views about being happy in our life with my spouse
18. My spouse and I have similar ideas about how marriage should be
19. My spouse and I have similar ideas about how roles should be in marriage
20. My spouse and I have similar values in trust.
21. I know exactly what my wife likes.
22. I know how my spouse wants to be taken care of when she/he sick.
23. I know my spouse's favourite food.
24. I can tell you what kind of stress my spouse is facing in her/his life.
25. I have knowledge of my spouse's inner world.
26. I know my spouse's basic anxieties.
27. I know what my spouse's current sources of stress are.
28. I know my spouse's hopes and wishes.
29. I know my spouse very well.
30. I know my spouse's friends and their social relationships.
31. I feel aggressive when I argue with my spouse.

32. When discussing with my spouse, I usually use expressions such as 'you always' or 'you never'.
33. I can use negative statements about my spouse's personality during our discussions.
34. I can use offensive expressions during our discussions.
35. I can insult my spouse during our discussions.
36. I can be humiliating when we discussions.
37. My discussion with my spouse is not calm.
38. I hate my spouse's way of open a subject.
39. Our discussions often occur suddenly.
40. We're just starting a discussion before I know what's going on.
41. When I talk to my spouse about something, my calm suddenly breaks.
42. When I argue with my spouse, I only go out and I don't say a word.
43. I mostly stay silent to calm the environment a little bit.
44. Sometimes I think it's good for me to leave home for a while.
45. I'd rather stay silent than discuss with my spouse.
46. Even if I'm right in the discussion, I stay silent to hurt my spouse.
47. When I discuss with my spouse, I stay silent because I am afraid of not being able to control my anger.
48. I feel right in our discussions.
49. I have nothing to do with what I've been accused of.
50. I'm not actually the one who's guilty about what I'm accused of.
51. I'm not the one who's wrong about problems at home.
52. I wouldn't hesitate to tell my spouse about her/his inadequacy.
53. When I discuss, I remind my spouse of her/his inadequacy.
54. I'm not afraid to tell my spouse about her/his incompetence.

The above dataset can be downloaded from

<http://archive.ics.uci.edu/ml/datasets/Divorce+Predictors+data+set>.

3.3 ALGORITHM USED

3.3.1 Loading the Dataset

```
import pandas as pd
import numpy as np

df = pd.read_csv(r"C:\Users\NERV\Desktop\ML PROJECT\divorce.csv",sep=";")
df
```

Fig: 3.1: Loading Dataset

The above code imports both pandas and NumPy and also loads our required dataset divorce.csv into the variable df from the file location C:\Users\NERV\Desktop\ML PROJECT.

Out[1]:

	Atr1	Atr2	Atr3	Atr4	Atr5	Atr6	Atr7	Atr8	Atr9	Atr10	...	Atr46	Atr47	Atr48	Atr49	Atr50	Atr51	Atr52	Atr53	Atr54	Class
0	2	2	4	1	0	0	0	0	0	0	...	2	1	3	3	3	2	3	2	1	1
1	4	4	4	4	4	0	0	4	4	4	...	2	2	3	4	4	4	4	2	2	1
2	2	2	2	2	1	3	2	1	1	2	...	3	2	3	1	1	1	2	2	2	1
3	3	2	3	2	3	3	3	3	3	3	...	2	2	3	3	3	3	2	2	2	1
4	2	2	1	1	1	1	0	0	0	0	...	2	1	2	3	2	2	2	1	0	1
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
165	0	0	0	0	0	0	0	0	0	0	...	1	0	4	1	1	4	2	2	2	0
166	0	0	0	0	0	0	0	0	0	0	...	4	1	2	2	2	2	3	2	2	0
167	1	1	0	0	0	0	0	0	0	1	...	3	0	2	0	1	1	3	0	0	0
168	0	0	0	0	0	0	0	0	0	0	...	3	3	2	2	3	2	4	3	1	0
169	0	0	0	0	0	0	0	1	0	0	...	3	4	4	0	1	3	3	3	1	0

170 rows × 55 columns

Fig 3.2: Desired Dataset

As observable, our dataset contains of categorical bivariate variables spanning over 170 rows and 55 columns. Our goal is to predict over the Class attribute using decision tree algorithm.

3.3.2 Extracting the target variable from the dataset

The below code stores our predict class into the y variable and stores the dataset without the predicting Class inside of the X variable.

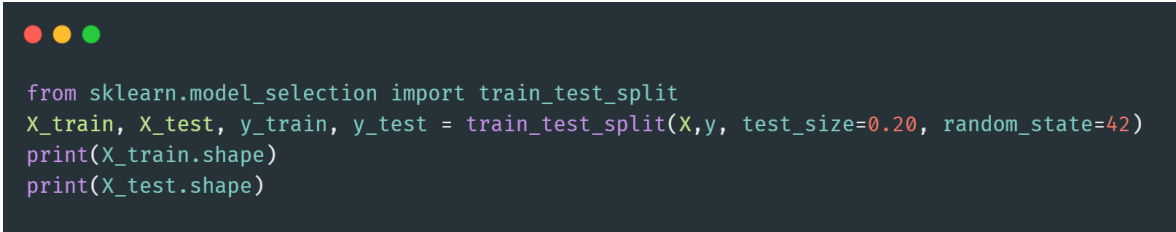


```
y= df['Class']  
X=df.drop('Class',axis=1)
```

Fig: 3.3: Extracting class attribute into a separate variable and dropping it from the table

3.3.3 Splitting the data into training and testing sets

Now that we have extracted the data attributes and corresponding labels, we will split them to form train and test datasets. For this purpose, we will use the scikit-learn's 'train_test_split' function, which takes in the attributes and labels as inputs and produces the train and test sets.



```
from sklearn.model_selection import train_test_split  
X_train, X_test, y_train, y_test = train_test_split(X,y, test_size=0.20, random_state=42)  
print(X_train.shape)  
print(X_test.shape)
```

Fig: 3.4: Using train_test_split to split train and test datasets

3.3.4 Fitting the model in Decision Tree Classifier

Since, this is a classification problem, we will import the DecisionTreeClassifier

function from the sklearn library. We'll also import accuracy_score metrics and tree from sklearn library. We'll fit our training set in the DecisionTreeClassifier () function.



```
import pandas as pd
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score
from sklearn import tree

clf = DecisionTreeClassifier()
clf = clf.fit(X_train,y_train)
```

Fig:3.5: Fitting our model

3.3.5 Building and Saving our tree as a png file

We can use Scikit-learn's export_graphviz function to display the tree within a Jupyter notebook. For plotting trees, we also need to install Graphviz and pydotplus.

export_graphviz function converts decision tree classifier into dot file and pydotplus convert this dot file to png or displayable form on Jupyter Notebook.

```

import graphviz
from six import StringIO
from IPython.display import Image
from sklearn.tree import export_graphviz
import pydotplus
dot_data = StringIO()
export_graphviz(clf, out_file=dot_data,
                filled=True, rounded=True,
                special_characters=True, feature_names = list(X.columns), class_names=['0', '1'])
graph = pydotplus.graph_from_dot_data(dot_data.getvalue())
graph.write_png('divorce.png')
Image(graph.create_png())

```

Fig: 3.6: Plotting and saving the tree

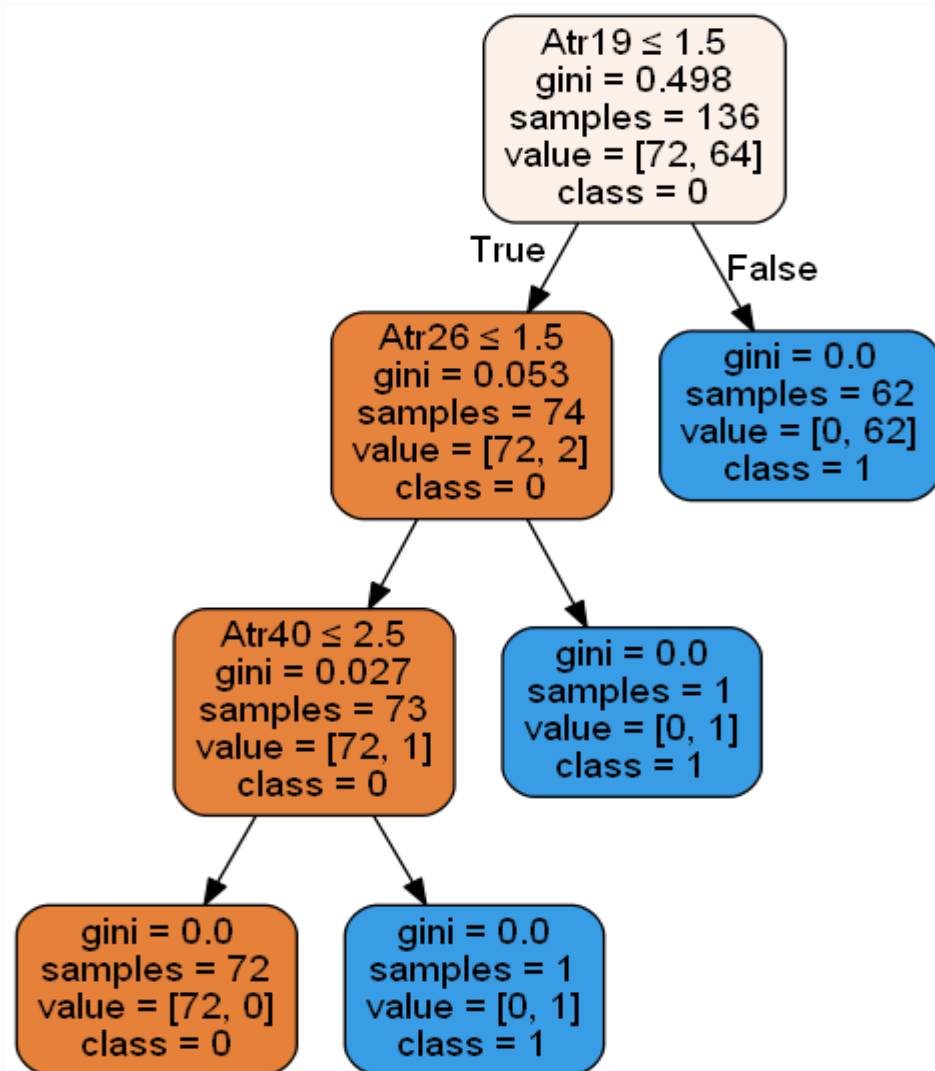


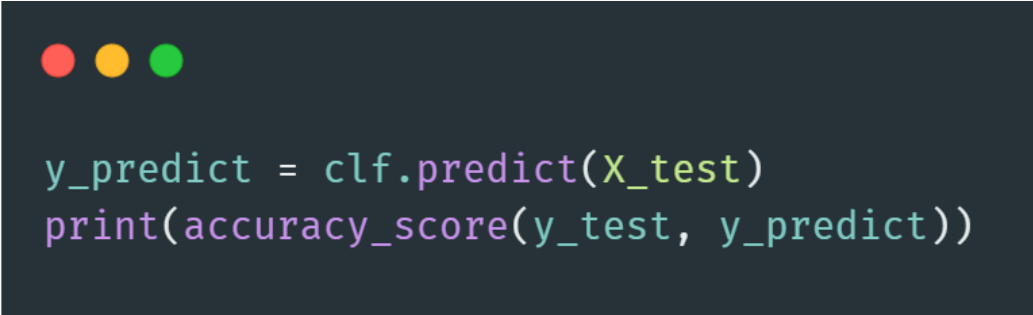
Fig: 3.7: Decision Tree

CHAPTER 4

PERFORMANCE METRICS AND RESULTS

4.1 PERFORMANCE METRICS

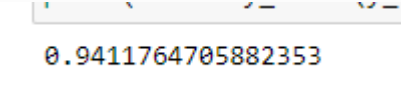
4.1.1 Accuracy Score

A dark-themed terminal window with three colored window control buttons (red, yellow, green) in the top-left corner. It contains two lines of Python code: `y_predict = clf.predict(X_test)` and `print(accuracy_score(y_test, y_predict))`.

```
y_predict = clf.predict(X_test)
print(accuracy_score(y_test, y_predict))
```

Fig: 4.1: Checking the accuracy

The above code predicts a series of our test data through our model and gives us an accuracy score.

A light-themed terminal window showing the output of the accuracy score calculation: `0.9411764705882353`.

```
0.9411764705882353
```

Fig: 4.2 Accuracy of our model

4.1.2 Confusion Matrix

We will now import `confusion_matrix` from `sklearn.metrics` to use it on our model.



```
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_predict)
cm
```

Fig: 4.3: Confusion Matrix code

```
Out[8]: array([[13,  1],
               [ 1, 19]], dtype=int64)
```

Fig: 4.4: Confusion Matrix of our model

Our confusion matrix fits in line well with our accuracy score.

The first row is about the not-divorced-predictions: 13 couples were correctly classified as not divorced (called true negatives) and 1 were wrongly classified as not divorced (false positives).

The second row is about the divorced-predictions: 1 couple was wrongly classified as divorced (false negatives) and 19 were correctly classified as divorced (true positives).

A confusion matrix gives you a lot of information about how well your model does, but there's a way to get even more, like computing the classifiers precision.

4.2 RESULTS

In this project, we have covered a lot of details about Decision Tree; It's working, attribute selection measures, decision tree model building, visualization and evaluation on divorce prediction dataset using Python Scikit-learn package.

We've seen that Decision tree algorithms outperforms others in terms of accuracy, time and precision. It quite relies on the algorithm used for recommendation to find interesting resources.

However, Decision trees are of unstable nature, they can often appear to be overfitting(memorizing) the training dataset and special care should be taken to avoid it as it leads to bad predictions.

```
In [9]: y_predict = clf.predict(X_train)
        print(accuracy_score(y_train, y_predict))

1.0

In [10]: clf.score(X,y)

Out[10]: 0.9882352941176471
```

Fig: 4.5: Score

However, with smaller datasets like ours, it can also be a false alarm because there just isn't enough data to make more complex and pronounced predictions.

CHAPTER 5

SUMMARY AND CONCLUSIONS

To conclude, we successfully demonstrated the usage of Decision Tree on Divorce Predictors Dataset. We build our model with 94% accuracy and demonstrated how useful and easy to use Decision Trees can be. Divorce prediction can prove useful in making informed decisions about the future and also in studies relating to couples and to infer patterns in behaviours of the divorced couples. The study concludes that Decision Trees are easy to use, understand, can handle both categorical and numerical data while having high precision and being resistant to outliers and while they are susceptible to overfitting (memorizing the training dataset), with a little bit of care, anyone can pick up and utilize one of the best machine learning algorithms available.

Decision trees algorithms outperform others in terms of accuracy, time and precision. It relies quite heavily on the algorithm used for recommendation to find interesting resources making them not require as much pre-processing. In conclusion, we successfully developed a decision tree model for our divorce prediction dataset with high precision results.

5.1 FUTURE WORK

Project can be extended to various industry applications such as assessing future growth opportunities via predictions, and also be used in the use of demographic data to find prospective clients. They can help streamline a marketing budget and make informed decisions on the target market that the business is focused on. In the absence of decision trees, the business may spend its marketing market without a specific demographic in mind, which will affect its overall revenues. This study in particular can also be expanded in relation to couples and to infer patterns in behaviours of the divorced couples at large. It can also help in determining better compatibility in marriage and drive more informed decisions about marriage life. The study can also be expanded to a real-time marriage legal consultation that works automatically with better results than other options available.

REFERENCES

PROJECT:

- Yöntem, Mustafa & Adem, Kemal & İlhan, Tahsin & Kılıçarslan, Serhat. (2019). DIVORCE PREDICTION USING CORRELATION BASED FEATURE SELECTION AND ARTIFICIAL NEURAL NETWORKS
- C. Wang, D. Chen, Y. Hu, Y. Ceng, J. Chen and H. Li, "Automatic Dialogue System of Marriage Law Based on the Parallel C4.5 Decision Tree," in IEEE Access, vol. 8, pp. 36061-36069, 2020, doi: 10.1109/ACCESS.2020.2972586.
- A. Sharma, A. S. Chudhey and M. Singh, "Divorce case prediction using Machine learning algorithms," 2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS), 2021, pp. 214-219, doi: 10.1109/ICAIS50930.2021.9395860.
- www.google.com
- www.geeksforgeeks.com
- Cognibot lectures & recorded sessions ([Dashboard \(cognibot.ml\)](http://Dashboard(cognibot.ml)))

DATASET:

1. Dr. Mustafa Kemal Yöntem, Nevşehir Hacı Bektaş Veli University, Faculty of Education, Department of Educational Sciences, muskemtem@hotmail.com
2. Dr. Kemal ADEM, Aksaray University, Faculty of Economics and Administrative Sciences, Department of Management Information Systems, kemaladem@gmail.com
3. Prof. Dr. Tahsin İlhan, Tokat GAZİOSMANPAŞA University, Faculty of Education, Department of Educational Sciences, tahsinilhan73@gmail.com
4. Lecturer Serhat Kılıçarslan, Tokat GAZİOSMANPAŞA University, Rectorate, Department of Informatics, serhatklc@gmail.com

APPENDIX

SCREENSHOTS

DATA LOADING AND SPLITTING

Loading Our Dataset

```
In [1]: import pandas as pd
import numpy as np

df = pd.read_csv(r"C:\Users\NERV\Desktop\ML PROJECT\divorce.csv", sep=";")
df
```

```
Out[1]:
```

	Atr1	Atr2	Atr3	Atr4	Atr5	Atr6	Atr7	Atr8	Atr9	Atr10	...	Atr46	Atr47	Atr48	Atr49	Atr50	Atr51	Atr52	Atr53	Atr54	Class
0	2	2	4	1	0	0	0	0	0	0	...	2	1	3	3	3	2	3	2	1	1
1	4	4	4	4	4	0	0	4	4	4	...	2	2	3	4	4	4	4	2	2	1
2	2	2	2	2	1	3	2	1	1	2	...	3	2	3	1	1	1	2	2	2	1
3	3	2	3	2	3	3	3	3	3	3	...	2	2	3	3	3	3	2	2	2	1
4	2	2	1	1	1	1	0	0	0	0	...	2	1	2	3	2	2	2	1	0	1
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
165	0	0	0	0	0	0	0	0	0	0	...	1	0	4	1	1	4	2	2	2	0
166	0	0	0	0	0	0	0	0	0	0	...	4	1	2	2	2	2	3	2	2	0
167	1	1	0	0	0	0	0	0	0	1	...	3	0	2	0	1	1	3	0	0	0
168	0	0	0	0	0	0	0	0	0	0	...	3	3	2	2	3	2	4	3	1	0
169	0	0	0	0	0	0	0	1	0	0	...	3	4	4	0	1	3	3	3	1	0

170 rows x 55 columns

```
In [2]: #extracting the target variable from the dataset
y= df['Class']
X=df.drop('Class',axis=1)

In [3]: #splitting the data into 80-20 test-train sets to measure performance

from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X,y, test_size=0.20, random_state=42)
print(X_train.shape)
print(X_test.shape)
```

```
(136, 54)
(34, 54)
```

FITTING THE MODEL

Fitting the model in Decision tree Classifier

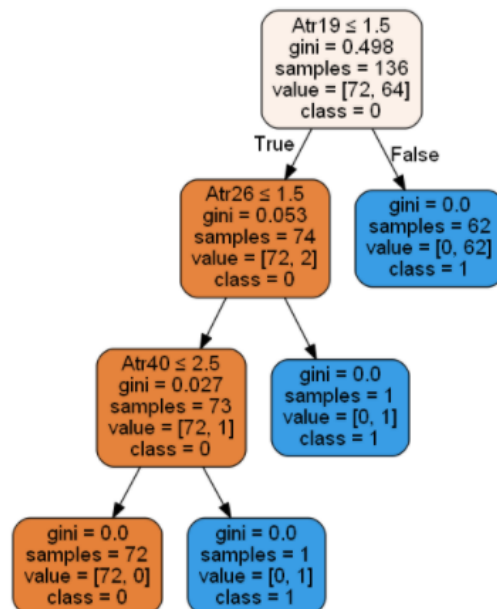
```
In [4]: import pandas as pd
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score
from sklearn import tree
```

```
clf = DecisionTreeClassifier()
clf = clf.fit(X_train,y_train)
```

```
In [5]: #beautiful tree
```

```
import graphviz
from six import StringIO
from IPython.display import Image
from sklearn.tree import export_graphviz
import pydotplus
dot_data = StringIO()
export_graphviz(clf, out_file=dot_data,
                filled=True, rounded=True,
                special_characters=True, feature_names = list(X.columns), class_names=['0','1'])
graph = pydotplus.graph_from_dot_data(dot_data.getvalue())
graph.write_png('divorce.png')
Image(graph.create_png())
```

Out[5]:



PERFORMANCE OF OUR MODEL

```
In [6]: """(same code but doesn't make a png file)
```

```
import graphviz

dot_data = tree.export_graphviz(clf, out_file=None, filled=True,
rounded=True, special_characters=True, feature_names=list(X.columns), class_names=['0', '1'])

graph = graphviz.Source(dot_data)
graph
"""
```

```
Out[6]: "(same code but doesn't make a png file)\n\nimport graphviz \n\ndot_data = tree.export_graphviz(clf, out_file=None, filled=True,\nrounded=True, special_characters=True, feature_names=list(X.columns), class_names=['0', '1']) \n\ngraph = graphviz.Source(dot_data)\ngraph\n"
```

Accuracy of our Model

```
In [7]: #validation accuracy
y_predict = clf.predict(X_test)
print(accuracy_score(y_test, y_predict))

0.9411764705882353
```

```
In [8]: from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_predict)
cm
```

```
Out[8]: array([[13,  1],
               [ 1, 19]], dtype=int64)
```

```
In [9]: y_predict = clf.predict(X_train)
print(accuracy_score(y_train, y_predict))

1.0
```

```
In [10]: clf.score(X,y)
```

```
Out[10]: 0.9882352941176471
```