Министерство науки и высшего образования Российской Федерации
Калужский филиал
федерального государственного бюджетного образовательного
учреждения высшего образования
**«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»**
(КФ МГТУ им. Н.Э. Баумана)

И.И. Ерохин

КЛАССИФИКАЦИЯ С ПОМОЩЬЮ ДЕРЕВЬЕВ РЕШЕНИЙ.

Методические указания к выполнению лабораторной работы

по курсу «Технологии анализа данных»

Калуга – 2020

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ

Настоящие методические указания составлены в соответствии с программой проведения лабораторных работ по курсу «Технологии анализа данных» на кафедре «Программное обеспечение ЭВМ, информационные технологии» факультета «Информатика и управление» Калужского филиала МГТУ им. Н.Э. Баумана.

Методические указания, ориентированные на студентов 4-го курса направления подготовки 09.03.04 «Программная инженерия», содержат краткую теоретическую часть, описывающую область применения деревьев решений.

Методические указания составлены в расчете на всестороннее ознакомление студентов с основами работы с методами классификации с использованием деревьев решений.

ЦЕЛЬ И ЗАДАЧИ РАБОТЫ, ТРЕБОВАНИЯ К РЕЗУЛЬТАТАМ ЕЕ ВЫПОЛНЕНИЯ

Целью выполнения лабораторной работы является формирование практических навыков работы с деревьями принятия решений.

Основными задачами выполнения лабораторной работы являются:

1. Ознакомиться с работой деревьев принятия решений.

Результатами работы являются:

1. Построенное дерево решений.
2. Подготовленный отчет.

КРАТКАЯ ХАРАКТЕРИСТИКА ОБЪЕКТА ИЗУЧЕНИЯ, ИССЛЕДОВАНИЯ

Деревья решений

Своевременная разработка и принятие правильного решения - это одна из главных задач работы управленческого персонала организации, т.к. необдуманное решение может дорого обойтись компании. Зачастую на практике результат одного решения заставляет нас принимать следующее решение и т. д. Когда же нужно принять несколько решений в условиях неопределенности, когда каждое решение зависит от исхода предыдущего, то применяют схему, называемую деревом решений.

Дерево решений — это графическое изображение процесса принятия решений, в котором отражены альтернативные решения, соответствующие вероятности, и выигрыши для любых комбинаций альтернатив. Структура показана на рисунке 1.

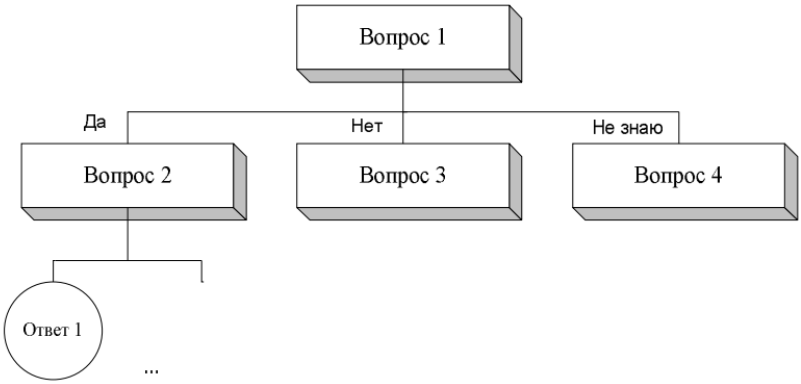

Рис. 1. Структура дерева решений

Модели на основе деревьев принятия решений обладают многими преимуществами. Они понятны и легко интерпретируемы, а процесс, на основе которого они достигают прогноза, абсолютно прозрачен. Модели на основе деревьев принятия решений легко справляются с сочетанием численных (количество ног) и категорийных (вкусный/ не

вкусный) признаков и даже могут классифицировать данные, для которых атрибуты отсутствуют.

Как строится дерево решений

Здесь можно вспомнить игру "20 вопросов", которая часто упоминается во введении в деревья решений. Наверняка каждый в нее играл. Один человек загадывает знаменитость, а второй пытается отгадать, задавая только вопросы, на которые можно ответить "Да" или "Нет" (опустим варианты "не знаю" и "не могу сказать"). Какой вопрос отгадывающий задаст первым делом? Конечно, такой, который сильнее всего уменьшит количество оставшихся вариантов. К примеру, вопрос "Это Анджелина Джоли?" в случае отрицательного ответа оставит более 7 миллиардов вариантов для дальнейшего перебора (конечно, поменьше, не каждый человек – знаменитость, но все равно немало), а вот вопрос "Это женщина?" отсечет уже около половины знаменитостей. То есть, признак "пол" намного лучше разделяет выборку людей, чем признак "это Анджелина Джоли", "национальность-испанец" или "любит футбол". Это интуитивно соответствует понятию прироста информации, основанного на энтропии.

Энтропия

Энтропия Шеннона определяется для системы с N возможными состояниями следующим образом:

$$S = -\sum_{i=1}^{N} p_i \log_2 p_i$$

где p_i – вероятности нахождения системы в i-ом состоянии. Это очень важное понятие, используемое в физике, теории информации и других областях. Опуская предпосылки введения (комбинаторные и теоретико-информационные) этого понятия, отметим, что, интуитивно, энтропия соответствует степени хаоса в системе. Чем выше энтропия, тем менее упорядочена система и наоборот. Это

поможет нам формализовать "эффективное разделение выборки", про которое мы говорили в контексте игры "20 вопросов".

Области применения дерева решений

Дерево решений является прекрасным инструментом в системах поддержки принятия решений, интеллектуального анализа данных (Data Mining). В областях, где высока цена ошибки, они послужат отличным подспорьем аналитика или руководителя.

Дерево решений успешно применяется для решения практических задач в следующих областях:

- Банковское дело. Оценка кредитоспособности клиентов банка при выдаче кредитов.
- Промышленность. Контроль качества продукции (выявление дефектов), испытания без разрушений (например, проверка качества сварки) и т.д.
- Медицина. Диагностика различных заболеваний.
- Молекулярная биология. Анализ строения аминокислот.

ОБРАЗЕЦ ВЫПОЛНЕНИЯ ЗАДАНИЯ

```
#Подключение необходимых библиотек:

import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline

#Игнорирование предупреждений
import warnings
warnings.filterwarnings("ignore")

#Считывание данных из csv-файла

from sklearn import tree

df = pd.read_csv('iris_df.csv')
df.columns = ['X1', 'X2', 'X3', 'X4', 'Y']
df.head()

#Кодируем категориальные признаки с помощью техники One-Hot-Encoding.
#df = pd.concat([df, pd.get_dummies(df['Y'],prefix="Y")],axis=1)
#df.drop(['Y'],axis=1, inplace=True)
#df.head()

from sklearn.model_selection import train_test_split
decision = tree.DecisionTreeClassifier()

X = df.values[:, 0:4]
Y = df.values[:, 4]
trainX, testX, trainY, testY = train_test_split(X, Y, test_size=0.3

#Обучаем дерево
decision.fit(trainX, trainY)
print('Accuracy: \n', decision.score(testX, testY))

#Визуализируем дерево

from sklearn.externals.six import StringIO
from IPython.display import Image
import pydotplus as pydot

dot_data = StringIO()
tree.export_graphviz(decision, feature_names=['X1', 'X2', 'X3',
'X4'], out_file=dot_data,filled=True)
graph = pydot.graph_from_dot_data(dot_data.getvalue())
Image(graph.create_png())
```

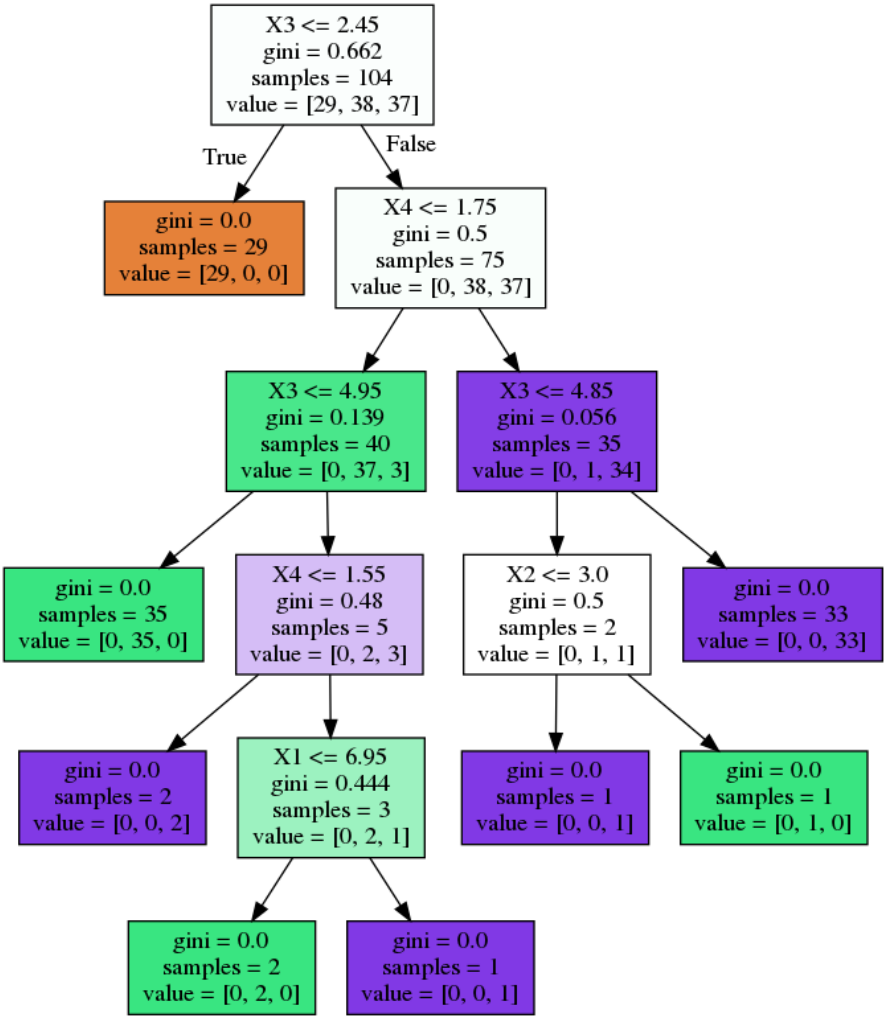

Рис. 2. Дерево принятия решений

ЗАДАНИЕ НА ЛАБОРАТОРНУЮ РАБОТУ

Создать дерево принятия решений согласно варианту, полученному у преподавателя.

ТРЕБОВАНИЯ К РЕАЛИЗАЦИИ

В качестве результата работы необходимо построить дерево принятия решений. По завершении готовится отчёт.

ВАРИАНТЫ ЗАДАНИЙ

Вариант 1

Считать данные из файла iris_df.csv в структуру DataFrame. Построить деревья решений с различной глубиной (не менее двух). В качестве выходного используйте поле «Класс».

Вариант 2

Считать тренировочные данные из файла titanic_train.csv в структуру DataFrame. Считать тестовые данные из файла titanic_test.csv в структуру DataFrame. Построить деревья решений с различной глубиной (не менее двух). В качестве выходного используйте поле «Age».

Вариант 3

Считать данные из файла titanic_data.csv в структуру DataFrame. Построить деревья решений с различной глубиной (не менее двух). В качестве выходного используйте поле «Pclass».

Вариант 4

Загрузить данные wine из библиотеки sklearn.datasets. Построить деревья решений с различной глубиной (не менее двух). В качестве выходного используйте поле «class_1».

Вариант 5

Считать данные из файла iris_df.csv в структуру DataFrame. Построить деревья решений с различной глубиной (не менее двух). В качестве выходного используйте поле «Sepal width».

Вариант 6

Считать тренировочные данные из файла titanic_train.csv в структуру DataFrame. Считать тестовые данные из файла titanic_test.csv в структуру DataFrame. Построить деревья решений с различной глубиной (не менее двух). В качестве выходного используйте поле «Sex».

Вариант 7

Считать данные из файла titanic_data.csv в структуру DataFrame. Построить деревья решений с различной глубиной (не менее двух). В качестве выходного используйте поле «Age».

Вариант 8

Загрузить данные wine из библиотеки sklearn.datasets. Построить деревья решений с различной глубиной (не менее двух). В качестве выходного используйте поле «class_3».

Вариант 9

Считать данные из файла iris_df.csv в структуру DataFrame. Построить деревья решений с различной глубиной (не менее двух). В качестве выходного используйте поле «Sepal lenght».

Вариант 10

Считать тренировочные данные из файла titanic_train.csv в структуру DataFrame. Считать тестовые данные из файла titanic_test.csv в структуру DataFrame. Построить деревья решений с различной глубиной (не менее двух). В качестве выходного используйте поле «pclass».

Вариант 11

Считать данные из файла titanic_data.csv в структуру DataFrame. Построить деревья решений с различной глубиной (не менее двух). В качестве выходного используйте поле «Sex».

Вариант 12

Загрузить данные wine из библиотеки sklearn.datasets. Построить деревья решений с различной глубиной (не менее двух). В качестве выходного используйте поле «class_2».

Вариант 13

Считать данные из файла iris_df.csv в структуру DataFrame. Построить деревья решений с различной глубиной (не менее двух). В качестве выходного используйте поле «Petal width».

Вариант 14

Считать тренировочные данные из файла titanic_train.csv в структуру DataFrame. Считать тестовые данные из файла titanic_test.csv в структуру DataFrame. Построить деревья решений с различной глубиной (не менее двух). В качестве выходного используйте поле «boat».

Вариант 15

Считать данные из файла titanic_data.csv в структуру DataFrame. Построить деревья решений с различной глубиной (не менее двух). В качестве выходного используйте поле «Embarked».

КОНТРОЛЬНЫЕ ВОПРОСЫ И ЗАДАНИЯ

1. Дайте определение дерева решений.
2. Опишите назначение деревья решений
3. Расскажите как строится дерево решений
4. Приведите пример использования деревьев решений при проведении анализа данных
5. Сформулируйте параметры, влияющие на обучения дерева решений

ФОРМА ОТЧЕТА ПО ЛАБОРАТОРНОЙ РАБОТЕ

На выполнение лабораторной работы отводится 1 занятие (2 академических часа: 1 час на выполнение и сдачу лабораторной работы и 1 час на подготовку отчета).

Отчет на защиту предоставляется в печатном виде.

Структура отчета (на отдельном листе(-ах)): титульный лист, формулировка задания, описание процесса выполнения лабораторной работы, результаты выполнения работы, выводы.

ОСНОВНАЯ ЛИТЕРАТУРА

1. Маккинли, Уэс Python и анализ данных / Пер. с англ. Слинкин А.А. - М.: ДМК Пресс, 2015. - 482 с.:ил.
2. Грас, Дж. Data Science. Наука о данных с нуля / Пер. с англ. - СПБ.: БХВ -Петербург, 2017. - 336с.: ил.

ДОПОЛНИТЕЛЬНАЯ ЛИТЕРАТУРА

3. Henley, A.J. Learn Data Analysis with Python: Lessons in Coding / A.J. Henley, Dave Wolf ISBN 978-1-4842-3486-0

Электронные ресурсы:

4. Научная электронная библиотека http://eLIBRARY.RU
5. Электронно-библиотечная система http://e.lanbook.com