МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ

Навчально-науковий інститут інформаційних технологій

Кафедра інженерії програмного забезпечення

**КОМП’ЮТЕРНИЙ ПРАКТИКУМ 2.**

ПРОГНОЗУВАННЯ НА ОСНОВІ КЛАСИФІКАТОРА  
ДЕРЕВА РІШЕНЬ

Підготував: студент групи ПДМ-51

Гапей Максим Юрійович

Перевірив: викладач

Садовенко Володимир Сергійович

Київ 2024

**Мета роботи:** потрібно створити класифікатор дерева рішень, який працюватиме з набором даних про пасажирів, які були на борту пасажирського лайнера «Титанік» під час його сумнозвісного рейсу.

**Частина 1 Створення класифікатора дерева рішень**

**Крок 1. Створіть фрейм даних.**

а) Імпортуйте pandas та файл csv.

import pandas as pd

training = pd.read\_csv("titanic\_train.csv")

б) Перевірте імпорт і прогляньте дані.

training.info()

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 891 entries, 0 to 890

Data columns (total 12 columns):

# Column Non-Null Count Dtype

--- ------ -------------- -----

0 PassengerId 891 non-null int64

1 Survived 891 non-null int64

2 Pclass 891 non-null int64

3 Name 891 non-null object

4 Sex 891 non-null object

5 Age 714 non-null float64

6 SibSp 891 non-null int64

7 Parch 891 non-null int64

8 Ticket 891 non-null object

9 Fare 891 non-null float64

10 Cabin 204 non-null object

11 Embarked 889 non-null object

dtypes: float64(2), int64(5), object(5)

memory usage: 83.7+ KB

**Крок 2. Підготуйте дані для побудови моделі дерева рішень.**

а) Замініть рядкові дані числовими мітками.

training["Sex"] = training["Sex"].apply(lambda toLabel: 0 if toLabel == 'male' else 1)

б) Переконайтеся, що змінна Gender була змінена.

training.info()

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 891 entries, 0 to 890

Data columns (total 12 columns):

# Column Non-Null Count Dtype

--- ------ -------------- -----

0 PassengerId 891 non-null int64

1 Survived 891 non-null int64

2 Pclass 891 non-null int64

3 Name 891 non-null object

4 Sex 891 non-null int64

5 Age 714 non-null float64

6 SibSp 891 non-null int64

7 Parch 891 non-null int64

8 Ticket 891 non-null object

9 Fare 891 non-null float64

10 Cabin 204 non-null object

11 Embarked 889 non-null object

dtypes: float64(2), int64(6), object(4)

memory usage: 83.7+ KB

в) Зверніться до відсутніх значень у наборі даних.

training["Age"].fillna(training["Age"].mean(), inplace=True)

*Наведіть число відсутніх значень для атрибуту віку:*

Відповідь: 891-741=150, де 891 — усього, 741 — присутньо.

г) Переконайтеся, що значення були замінені.

training.info()

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 891 entries, 0 to 890

Data columns (total 12 columns):

# Column Non-Null Count Dtype

--- ------ -------------- -----

0 PassengerId 891 non-null int64

1 Survived 891 non-null int64

2 Pclass 891 non-null int64

3 Name 891 non-null object

4 Sex 891 non-null int64

5 Age 891 non-null float64

6 SibSp 891 non-null int64

7 Parch 891 non-null int64

8 Ticket 891 non-null object

9 Fare 891 non-null float64

10 Cabin 204 non-null object

11 Embarked 889 non-null object

dtypes: float64(2), int64(6), object(4)

memory usage: 83.7+ KB

*Яке значення було використано для заміни відсутніх значень віку?*

Відповідь: training["Age"].mean() # 29.69911764705882 ~ 30

**Крок 3. Навчіть і оцініть модель класифікатора дерева рішень.**

а) Створіть об'єкт масиву зі змінною, яка буде цільовою для моделі.

Training\_target = training["Survived"].values

б) Створіть масив значень, які будуть вхідними для моделі.

columns = ["Fare", "Pclass", "Sex", "Age", "SibSp"]

Training\_input = training[list(columns)].values

в) Створіть модель класифікатора дерева рішень.

from sklearn import tree

import six

import sys

sys.modules['sklearn.externals.six'] = six

clf\_train = tree.DecisionTreeClassifier(criterion="entropy", max\_depth=3)

clf\_train = clf\_train.fit(Training\_input, Training\_target)

г) Оцініть модель.

clf\_train.score(Training\_input, Training\_target)

**Крок 4. Візуалізуйте дерево рішень.**

а) Створіть вихідний проміжний файл.

from sklearn.externals.six import StringIO

б) Встановіть Graphviz.

with open("titanic.dot", 'w') as f:

f = tree.export\_graphviz(clf\_train, out\_file=f, feature\_names=columns)

в) Перетворіть проміжний файл у графічний.

digraph Tree {

node [shape=box, fontname="helvetica"] ;

edge [fontname="helvetica"] ;

0 [label="Sex <= 0.5\nentropy = 0.961\nsamples = 891\nvalue = [549, 342]"] ;

1 [label="Fare <= 26.269\nentropy = 0.699\nsamples = 577\nvalue = [468, 109]"] ;

0 -> 1 [labeldistance=2.5, labelangle=45, headlabel="True"] ;

2 [label="Age <= 13.5\nentropy = 0.558\nsamples = 415\nvalue = [361, 54]"] ;

1 -> 2 ;

3 [label="entropy = 0.567\nsamples = 15\nvalue = [2, 13]"] ;

2 -> 3 ;

4 [label="entropy = 0.477\nsamples = 400\nvalue = [359, 41]"] ;

2 -> 4 ;

5 [label="SibSp <= 2.5\nentropy = 0.924\nsamples = 162\nvalue = [107, 55]"] ;

1 -> 5 ;

6 [label="entropy = 0.964\nsamples = 139\nvalue = [85, 54]"] ;

5 -> 6 ;

7 [label="entropy = 0.258\nsamples = 23\nvalue = [22, 1]"] ;

5 -> 7 ;

8 [label="Pclass <= 2.5\nentropy = 0.824\nsamples = 314\nvalue = [81, 233]"] ;

0 -> 8 [labeldistance=2.5, labelangle=-45, headlabel="False"] ;

9 [label="Fare <= 28.856\nentropy = 0.299\nsamples = 170\nvalue = [9, 161]"] ;

8 -> 9 ;

10 [label="entropy = 0.469\nsamples = 70\nvalue = [7, 63]"] ;

9 -> 10 ;

11 [label="entropy = 0.141\nsamples = 100\nvalue = [2, 98]"] ;

9 -> 11 ;

12 [label="Fare <= 23.35\nentropy = 1.0\nsamples = 144\nvalue = [72, 72]"] ;

8 -> 12 ;

13 [label="entropy = 0.977\nsamples = 117\nvalue = [48, 69]"] ;

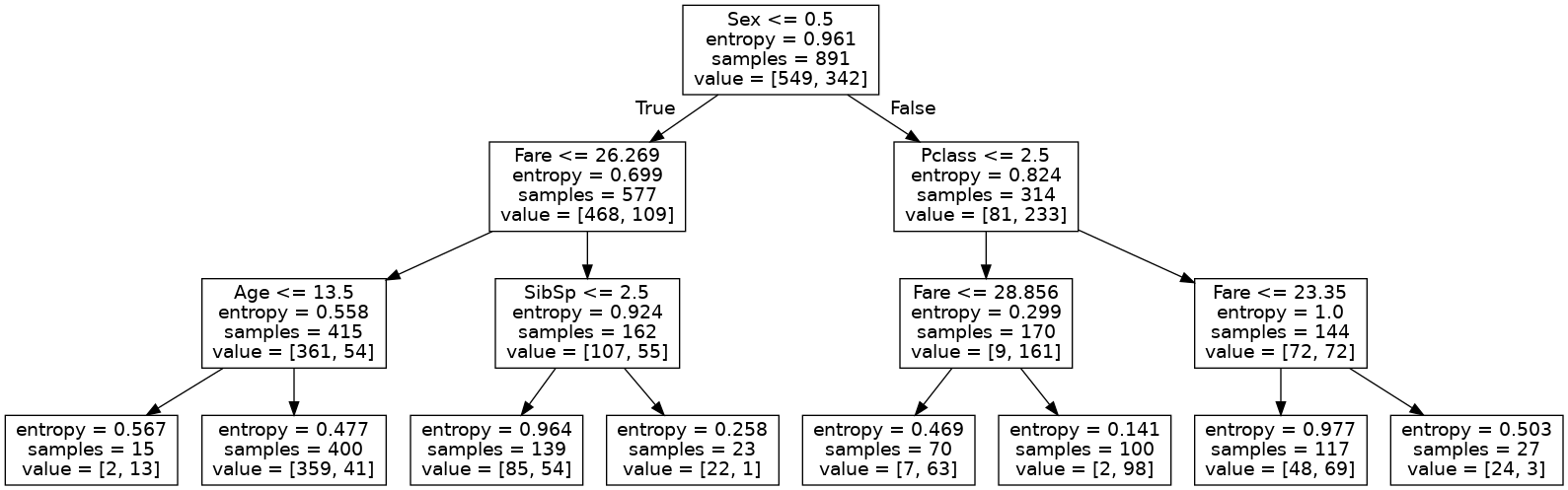
12 -> 13 ;

14 [label="entropy = 0.503\nsamples = 27\nvalue = [24, 3]"] ;

12 -> 14 ;

}

г) Покажіть зображення.



д) Інтерпретуйте дерево рішень.

*Виходячи з даних кореневого вузла, скільки спостережень складають*

*набір навчальних даних?*

Відповідь: 891

*Виходячи з даних наступного рівня, скільки пасажирів були чоловіками, а*

*скільки жінками?*

Відповідь: 549ч та 324ж

*Виходячи з даних к ореневого вузла, скільки людей з даної вибірки померли*

*та скільки вижили?*

Відповідь: 468+81=549 померли та 109+233=342 вижили

*Що характеризує групу, яка мала найбільше смертей за кількістю?*

Відповідь: Вік (діти у віці до 13 з половиною років)

*У якій групі вижило найбільше пасажирів?*

Відповідь: Pclass

**Частина 2 Застосування та оцінка моделі**

**класифікатора дерева рішень**

**Крок 5. Імпортуйте та підготуйте дані.**

а) Імпортуйте дані.

testing = pd.read\_csv("titanic\_test.csv")

testing.info()

# <class 'pandas.core.frame.DataFrame'>

# RangeIndex: 418 entries, 0 to 417

# Data columns (total 11 columns):

# # Column Non-Null Count Dtype

# --- ------ -------------- -----

# 0 PassengerId 418 non-null int64

# 1 Pclass 418 non-null int64

# 2 Name 418 non-null object

# 3 Sex 418 non-null object

# 4 Age 332 non-null float64

# 5 SibSp 418 non-null int64

# 6 Parch 418 non-null int64

# 7 Ticket 418 non-null object

# 8 Fare 417 non-null float64

# 9 Cabin 91 non-null object

# 10 Embarked 418 non-null object

# dtypes: float64(2), int64(4), object(5)

# memory usage: 36.1+ KB

*Скільки записів у наборі даних?*

Відповідь: 418

*Яких важливих змінних значень немає, скільки відсутніх?*

Відповідь: Age 418-332=86 (також Fare 418-417=1 та Cabin 418-91=327)

б) Використовуйте лямбда-вираз, щоб замінити значення "male" та

"female" на 0 для чоловіків і 1 для жінок.

testing["Sex"] = testing["Sex"].apply(lambda toLabel: 0 if toLabel == 'male' else 1)

в) Замініть пропущені значення віку на середнє значення віку.

testing["Age"].fillna(testing["Age"].mean(), inplace=True)

г) Переконайтеся, що значення були замінені.

testing.info()

# <class 'pandas.core.frame.DataFrame'>

# RangeIndex: 418 entries, 0 to 417

# Data columns (total 11 columns):

# # Column Non-Null Count Dtype

# --- ------ -------------- -----

# 0 PassengerId 418 non-null int64

# 1 Pclass 418 non-null int64

# 2 Name 418 non-null object

# 3 Sex 418 non-null int64

# 4 Age 418 non-null float64

# 5 SibSp 418 non-null int64

# 6 Parch 418 non-null int64

# 7 Ticket 418 non-null object

# 8 Fare 417 non-null float64

# 9 Cabin 91 non-null object

# 10 Embarked 418 non-null object

# dtypes: float64(2), int64(5), object(4)

# memory usage: 36.1+ KB

**Крок 6. Позначте набір даних тестування.**

а) Створіть масив вхідних змінних із набору даних тестування.

Testing\_input = testing[list(columns)].values

б) Застосуйте модель до набору даних тестування.

target\_labels = clf\_train.predict(Testing\_input)

target\_labels = pd.DataFrame({'Est\_Survival':target\_labels, 'Name':testing['Name']})

в) Оцініть точність передбачуваних міток.

import numpy as np

all\_data = pd.read\_csv("titanic\_all.csv")

testing\_results = pd.merge(target\_labels, all\_data[['Name','Survived']], on=['Name'])

acc = np.sum(testing\_results['Est\_Survival'] == testing\_results['Survived']) / float(len(testing\_results))

**Частина 3 Оцінка моделі класифікатора дерева рішень**

**Крок 7. Імпортуйте дані.**

all\_data = pd.read\_csv("titanic\_all.csv", usecols=['Survived','Pclass','Gender','Age','SibSp','Fare'])

all\_data.info()

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 1309 entries, 0 to 1308

Data columns (total 6 columns):

# Column Non-Null Count Dtype

--- ------ -------------- -----

0 Survived 1309 non-null int64

1 Pclass 1309 non-null int64

2 Sex 1309 non-null int64

3 Age 1046 non-null float64

4 SibSp 1309 non-null int64

5 Fare 1308 non-null float64

dtypes: float64(2), int64(4)

memory usage: 61.5 KB

*Скільки записів у наборі даних?*

Відповідь: 1309

*Яких важливих значень змінних немає, скільки всього відсутніх значень?*

Відповідь: Age 1309-1046=263

**Крок 8. Підготуйте дані.**

a) Видаліть рядки " male" і " female" і замініть їх на 0 і 1 відповідно.

all\_data["Sex"] = all\_data["Sex"].apply(lambda toLabel: 0 if toLabel == 'male' else 1)

б) Замініть пропущені значення віку на середнє значення віку всіх

пасажирів з набору даних.

all\_data["Age"].fillna(all\_data["Age"].mean(), inplace=True)

all\_data.info()

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 1309 entries, 0 to 1308

Data columns (total 6 columns):

# Column Non-Null Count Dtype

--- ------ -------------- -----

0 Survived 1309 non-null int64

1 Pclass 1309 non-null int64

2 Sex 1309 non-null int64

3 Age 1309 non-null float64

4 SibSp 1309 non-null int64

5 Fare 1308 non-null float64

dtypes: float64(2), int64(4)

memory usage: 61.5 KB

**Крок 9. Створіть вхідні та вихідні змінні для даних навчання та**

**тестування.**

а) Призначте вхідні та вихідні змінні та згенеруйте масиви.

from sklearn.cross\_validation import train\_test\_split

X = all\_data[list(columns)].values

y = all\_data["Survived"].values

X\_train,X\_test,y\_train,y\_test=train\_test\_split(X, y, test\_size=0.40, random\_state=0)

б) Навчіть модель і пристосуйте її до даних тестування.

clf\_train = tree.DecisionTreeClassifier(criterion="entropy", max\_depth=3)

clf\_train = clf\_train.fit(X\_train, y\_train)

в) Порівняйте моделі, оцінивши кожну.

train\_score = str(clf\_train.score(X\_train,y\_train))

test\_score = str(clf\_train.score(X\_test,y\_test))

print('Training score = '+ train\_score+' Testing score = '+test\_score)

*Який результат ви отримали?*

Відповідь: Training score = 99.77% Testing score = 99.86%

**Висновки:** Під час виконання практичної роботи, було ознайомлено з пакетами Python: pandas, numpy та csv. Завантажено та підготувано дані csv. Після чого створено та візуалізовано дерево рішень.