Lung Cancer Prediction

โดย

64010462 บุริศ เสรีวัฒนะ

64010659 ภัทราภรณ์ จันเดชา

64010801 วัทธิกร เจริญกัลป์

64010815 วิรุฬ สำเภาทอง

โครงงานนี้เป็นส่วนหนึ่งของการศึกษา

วิชา 01076032 ELEMENTARY DIFFERENTIAL EQUATIONS

AND LINEAR ALGEBRA

ปีการศึกษา 2565

บทคัดย่อ

โรคมะเร็งปอดนั้นเป็นโรคที่อันตรายถึงชีวิต และเป็นหนึ่งในสาเหตุของการเสียชีวิตอันดับต้น ๆ ของคนไทย แต่ทว่าอาการของโรคมะเร็งปอดมักจะไม่ค่อยแสดงอาการให้เราได้ทราบในระยะแรก แต่จะมีสัญญาณที่บ่งบอกถึงการเกิดโรคในระยะที่เป็นอันตรายถึงชีวิต ทางผู้จัดทำโครงงานในครั้งนี้มีวัตถุประสงค์เพื่อทำนายความเสี่ยงของการเกิดโรคมะเร็งปอด โดยตรวจสอบและวิเคราะห์รูปแบบของข้อมูลโดยแบ่งเป็น อายุ เพศ และอาการต่าง ๆ ที่เป็นปัจจัยเสี่ยงต่อการเกิดโรคมะเร็งปอด ยกตัวอย่าง เช่น การสูบบุหรี่ อาการภูมิแพ้ ความวิตกกังวล อากการไอ การดื่มแอลกอฮอล์ เป็นต้น ร่วมกับการใช้หลักการทางคณิตศาสตร์ คือ Pearson’s Similarity, Logistic Regression และ Confusion Matrix เพื่อนำรูปแบบข้อมูลข้างต้น ไปประมวลผลเข้าด้วยกัน ซึ่งเป็นหนึ่งในตัวช่วยที่สามารถตัดสินใจทำการตรวจหาโรคมะเร็งปอดและทำการรักษาได้อย่างทันเวลา

คำสำคัญ: โรคมะเร็งปอด, ทำนายความเสี่ยง, Logistic Regression

สารบัญ

บทที่ 1

บทนำ

1.1 ที่มาของโครงงาน

โรคมะเร็งปอด เป็นโรคที่สามารถทำอันตรายถึงชีวิต และเป็นหนึ่งในสาเหตุของการเสียชีวิตอันดับต้นๆ ของคนไทย แต่ทว่าอาการของโรคมะเร็งปอดมักจะไม่ค่อยแสดงอาการให้เราได้ทราบในระยะแรกแต่มักจะมีสัญญาณที่บ่งบอกถึงการ เกิดโรคในระยะที่เป็นอันตรายถึงชีวิต แต่ถ้าหากว่า เราสามารถวัดความเสี่ยงของการเกิดโรคมะเร็งปอด และ พบเจอในระยะต้น เราจะสามารถรักษาให้หาย และทำให้เราไม่เป็น อันตรายถึงชีวิต

คณะผู้จัดทำจึงได้ทำโครงงาน Lung Cancer Prediction ที่สามารถทำนายความเสี่ยงของการเกิดโรคมะเร็งปอด เพื่อเป็นหนึ่งในตัวช่วยที่สามารถตัดสินใจทำการตรวจหาโรคมะเร็งปอดและทำการรักษาได้อย่างทันท่วงที ก่อนที่จะสายเกินไป

1.2 จุดประสงค์โครงงาน

1.2.1. เพื่อให้ผู้ใช้งานสามารถทำนายความเสี่ยงที่อาจจะเกิดโรคมะเร็งปอดได้

1.2.2. เป็นตัวช่วยที่สามารถตัดสินใจทำการตรวจหาโรคมะเร็งปอดและรักษาได้อย่างทันเวลา

1.2.3. ได้เรียนรู้การใช้หลักการทางคณิตศาสตร์ ได้แก่ Pearson’s Similarity, Logistic Regression และConfusion Matrix

บทที่ 2

ภาพรวมการออกแบบระบบ

2.1 ภาพรวมขั้นตอนการทำงานของระบบ

ในการจัดทำโครงงานมีขั้นตอนการทำงานทั้งหมดดังนี้

2.1.1 หาข้อมูลที่เกี่ยวข้อง

ค้นหาข้อมูลที่ใช้สูตรทางคณิตศาศาสตร์ และข้อมูลที่จำเป็นในการทำโครงงาน

2.1.2 นำข้อมูลเข้าสู่โปรแกรม

เตรียมความพร้อมของโปรแกรมที่ใช้และนำข้อมูลเข้าสู่โปรแกรม

2.1.3 ตรวจสอบข้อมูล

ตรวจสอบข้อมูลที่ผิดปกติและทำการ Clean ข้อมูล

2.1.4 หาความสัมพันธ์ระหว่างปัจจัยต่างๆ

หาความสัมพันธ์ด้วย Pearson’s Similarity และสร้างกราฟเพื่อดูความสัมพันธ์

2.1.5 ประมวลผลข้อมูล

ประมวลผลข้อมูลสำหรับเข้าสู่ Model และแบ่งชุดข้อมูลสำหรับการทดลอง

2.1.6 สร้าง Model

นำข้อมูลทดลองเข้าสู่ Model Logical Regression และนำผลลัพธ์ที่ได้จากModel เข้าสู่ Confusion Matrix

2.2 รายละเอียดข้อมูลที่เกี่ยวข้อง

2.2.1 ความรู้ทั่วไปเกี่ยวกับโรคมะเร็งปอด

มะเร็งปอดมีต้นกำเนิดมาจากเซลล์เยื่อบุหลอดลมปอดที่ได้รับการระคายเคืองเป็นระยะเวลานานๆ จึงเรียกชื่อตามต้นกำเนิดของมะเร็งได้อีกชื่อหนึ่งว่า Bronchogenic Carcinoma ซึ่งอาจเกิดในบริเวณหลอดลมใหญ่ใกล้ขั้วปอด หรืออาจเกิดในหลอดลมแขนงเล็กๆ โดยมีปัจจัยเสี่ยงดังนี้

1. บุหรี่

บุหรี่เป็นสาเหตุของโรคมะเร็งปอดสูงถึงร้อยละ 80 – 90% การสูบบุหรี่ส่งผลต่อการเปลี่ยนแปลงของเซลล์หลอดลม ทำให้เกิดการกลายพันธุ์เป็นเซลล์มะเร็งได้ หากผู้ที่สูบบุหรี่จัดหยุดสูบบุหรี่จะมีความเสี่ยงต่อโรคมะเร็งปอดลดลงเรื่อย ๆ แต่กว่าจะลดลงจนเท่าคนที่ไม่สูบบุหรี่จะต้องใช้เวลากว่า 10 ปี

2. สารพิษและมลภาวะในสิ่งแวดล้อม เช่น

- สารแอสเบสตอส (asbestos) หรือแร่ใยหิน ที่ถูกนำมาใช้ในอุตสาหกรรมการผลิตวัสดุก่อสร้าง (กระเบื้องมุงหลังคา กระเบื้องแผ่นเรียบ ฝ้าเพดาน) อุตสาหกรรมการผลิตท่อน้ำซีเมนต์ กระเบื้องยางไวนิลปูพื้น ผ้าเบรก ฉนวนกันความร้อน และ อุตสาหกรรมสิ่งทอ แร่ใยหินเป็นสารที่ทำให้เกิดมะเร็งปอดได้ โดยผู้ที่ทำงานในโรงงานที่มีการใช้แร่ใยหินจะมีโอกาสเป็นมะเร็งปอดมากกว่าคนปกติถึง 7 เท่า

- ก๊าซเรดอน (radon) เป็นก๊าซสารกัมมันตภาพรังสีที่เกิดจากการสลายตัวของเรเดียมหรือยูเรเนียม พบได้ในอาคารหรือสิ่งก่อสร้างซึ่งมักมีดิน หิน หรือทรายที่มีแร่เรเดียมเจือปนมา

- สารเคมีอื่น ๆ เช่น สารหนู ถ่านหิน ที่ผู้ป่วยมักได้รับจากการประกอบอาชีพที่เกี่ยวกับโรงงานอุตสาหกรรม หรือสารพิษจากมลภาวะที่มาจากท่อไอเสียของยานพาหนะ

3. ฝุ่น PM 2.5

ฝุ่น PM 2.5 หรือฝุ่นละอองจิ๋วขนาดเล็กที่มีขนาดโมเลกุลเล็กเพียง 2.5 ไมครอน ที่ไม่สามารถมองไม่เห็นด้วยตาเปล่า โดยฝุ่น PM 2.5 จะเพิ่มความเสี่ยงในการเป็นมะเร็งปอดมากขึ้นถึง 1 – 1.4 เท่า โดยเมื่อฝุ่น PM 2.5 เข้าไปในปอดจะทำให้เกิดการอักเสบ และมีการกลายพันธุ์ของสารพันธุกรรม ซึ่งอาจทำให้เกิดมะเร็งปอดขึ้นได้

4. อายุ

อายุที่มากขึ้น อวัยวะรวมถึงเซลล์ต่าง ๆ ในร่างกายจะยิ่งทำงานเสื่อมสภาพลง โดยผู้ที่เสี่ยงจะเป็นมะเร็งปอดมากที่สุดจะอยู่ในช่วงอายุประมาณ 55 ปีขึ้นไป โดยเฉพาะอย่างยิ่งผู้ที่สูบบุหรี่

5. พันธุกรรม

ถ้ามีประวัติครอบครัวเป็นมะเร็งปอด จะมีความเสี่ยงต่อการเป็นโรคมะเร็งปอดเพิ่มขึ้นจากคนทั่วไป

6. ปัจจัยอื่นๆ

ปัจจัยอื่นๆ ที่มีความเสี่ยงต่อการเกิดโรคมะเร็งปอด เช่น อายุที่เพิ่มขึ้น การใช้ยาเสพติดบางประเภท เช่น โคเคน ภาวะขาดวิตามินเอ พันธุกรรม

2.2.2 หลักการของ Logistic Regression

การวิเคราะห์การถดถอยโลจีสติก (Logistic Regression Analysis) เป็นเทคนิคการวิเคราะห์ตัวแปรเชิงพหุที่มีวัตถุประสงค์เพื่อประมาณค่าหรือทำนายเหตุการณ์ที่สนใจว่าจะเกิดหรือไม่เกิดเหตุการณ์นั้นภายใต้อิทธิของตัวปัจจัย แบบจำลองโลจีสติกประกอบด้วยตัวแปรตาม (หรือตัวแปรเกณฑ์) ที่ต้องเป็นตัวแปรแบบทวินาม (Dichotomous Variable) กล่าวคือมีได้สองค่า เช่น “เกิด” กับ “ไม่เกิด” หรือ “เสี่ยง” กับ “ไม่เสี่ยง” เป็นต้น และตัวแปรอิสระ (หรือตัวแปรทำนาย) ที่อาจมีตัวเดียวหรือหลายตัวที่เป็นได้ทั้งตัวแปรเชิงกลุ่ม(Categorical Variable) หรือตัวแปรแบบต่อเนื่อง (Continuous Variable) การวิเคราะห์การถดถอยโลจีสติกเกี่ยวข้องกับทฤษฎีความน่าจะเป็นทวินามถูกเรียกว่า Binomial Logistic Regression ถ้าตัวแปรตามเป็นพหุนามจะเรียกว่า Multinomial Logistic Regression การถดถอยโลจีสติกจัดเป็นเครื่องมือวิเคราะห์ข้อมูลใน การศึกษาวิจัยที่มีวัตถุประสงค์เพื่อทำนายเหตุการณ์ หรือประเมินความเสี่ยง จึงมีการประยุกต์ใช้ในงานวิจัยหลากหลายสาขา ทั้งสาขาทางการแพทย์ วิศวกรรมศาสตร์ นิเวศวิทยา เศรษฐศาสตร์และสังคมศาสตร์

2.3 อธิบายขั้นตอนย่อยแต่ละขั้น

2.3.1 การหาข้อมูลที่เกี่ยวข้อง

เริ่มหาข้อมูลที่เกี่ยวข้องโดยเริ่มจากการหาข้อมูลเกี่ยวกับ Lung Cancer จาก Kaggle และศึกษาสูตรทางคณิตศาสตร์ที่จะใช้ในการสร้างโมเดลและการทำนาย โดยได้เลือก Pearson's Similarity และ Logistic Regression ในการคำนวณและสร้างโมเดล และ Confusion Matrix สำหรับการประเมินประสิทธิภาพในการทำนายของโมเดลที่สร้างขึ้น โดย Dataset มีจำนวนทั้งหมด 309 ชุดข้อมูล มี column เป็นข้อมูลเพศ อายุ และอาการต่าง ๆ ที่บ่งบอกถึงโรคมะเร็งปอด

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	GENDER	AGE	SMOKING	YELLOW_F	ANXIETY	PEER_PRE	CHRONIC	FATIGUE	ALLERGY	WHEEZIN(	ALCOHOL	COUGHIN(	SHORTNES	SWALLOW	CHEST PAI	LUNG_CANCER	
2	M	69	1	2	2	1	1	2	1	2	2	2	2	2	2	YES	
3	M	74	2	1	1	1	2	2	2	1	1	1	2	2	2	YES	
4	F	59	1	1	1	2	1	2	1	2	1	2	2	1	2	NO	
5	M	63	2	2	2	1	1	1	1	1	2	1	1	2	2	NO	
6	F	63	1	2	1	1	1	1	1	2	1	2	2	1	1	NO	
7	F	75	1	2	1	1	2	2	2	2	1	2	2	1	1	YES	
8	M	52	2	1	1	1	1	2	1	2	2	2	2	1	2	YES	
9	F	51	2	2	2	2	1	2	2	1	1	1	2	2	1	YES	
10	F	68	2	1	2	1	1	2	1	1	1	1	1	1	1	NO	
11	M	53	2	2	2	2	2	1	2	1	2	1	1	2	2	YES	
12	F	61	2	2	2	2	2	2	1	2	1	2	2	2	1	YES	
13	M	72	1	1	1	1	2	2	2	2	2	2	2	1	2	YES	

2.3.2 นำข้อมูลเข้าสู่โปรแกรม

ติดตั้ง Library ที่จำเป็นในการใช้เพื่อการคำนวณและสร้างโมเดล ได้แก่ pandas, NumPy, matplotlib, seaborn, scikit-learn, imbalanced-learn

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
from sklearn.preprocessing import LabelEncoder
import warnings
from imblearn.over_sampling import RandomOverSampler
from sklearn.model_selection import RandomizedSearchCV, train_test_split
from sklearn.preprocessing import StandardScaler
```

ตั้งค่า style ของ matplotlib และ Seaborn palette สำหรับใช้ในการสร้าง Figure

```
plt.style.use('fivethirtyeight')
colors=['#011f4b','#03396c','#005b96','#6497b1','#b3cde0']
sns.set_palette(sns.color_palette(colors))
```

หลังจากนั้นจึง ใช้ pandas อ่านข้อมูลไฟล์ csv เพื่อนำข้อมูลเข้าสู่โปรแกรม

```
## Read Dataset
df = pd.read_csv('survey_lung_cancer.csv')
```

2.3.3 ตรวจสอบข้อมูล

ตรวจสอบข้อมูลที่เป็นช่องว่างใน Dataset

```
# Check for null values
print(df.isnull().sum())
```

ตรวจสอบข้อมูลที่ซ้ำกันใน Dataset

```
# Check for duplicates in the dataset
print('Duplicated:',df.duplicated().sum())
```

แปลงข้อมูลที่เป็นข้อความเป็นตัวเลข เนื่องจากใน Dataset มี column “GENDER” และ “LUNG_CANCER” ที่มีข้อมูลเป็น M/F และ YES/NO ตามลำดับ

```
## Encoding LUNG_CANCER and GENDER column
encoder = LabelEncoder()
df['LUNG_CANCER']=encoder.fit_transform(df['LUNG_CANCER'])
df['GENDER']=encoder.fit_transform(df['GENDER'])
```

2.3.4 หาความสัมพันธ์ระหว่างปัจจัยต่างๆ

หาค่าความสัมพันธ์ระหว่างปัจจัยต่าง ๆ โดยใช้ library ของ seaborn และ matplotlib เพื่อสร้าง Heatmap และทำการคำนวณด้วยฟังก์ชั่นเพื่อหาค่าแบบ Pearson’s Similarity

```
## Heat map (Pearson's Similarity)
plt.figure(figsize=(15,15))
sns.heatmap(df.corr(),annot=True,linewidth=0.5,fmt='0.2f')
plt.show()
```

2.3.5 ประมวลผลข้อมูล

ทำการ drop column “LUNG_CANCER”

```
# Separating Independent and Dependent Feature
X = df.drop(['LUNG_CANCER'],axis=1)
y = df['LUNG_CANCER']
```

เปลี่ยนค่า 1 และ 2 ในข้อมูลเป็น 0 และ 1 ตามลำดับ เนื่องจากใน Dataset ใช้การแทน YES และ NO ด้วยตัวเลข 2 และ 1

```
# Changing values of columns from 2,1 to 1,0 (2/1 to 1/0 in each data)
for i in X.columns[2:]:
    temp=[]
    for j in X[i]:
        temp.append(j-1)
    X[i]=temp
```

ทำการ Oversampling ด้วยฟังก์ชั่นจาก imblearn เพื่อเพิ่มจำนวนข้อมูลให้มีข้อมูลในกลุ่มส่วนน้อยมีจำนวนใกล้เคียงกับกลุ่มส่วนมาก

```
# Oversampling of Minority Class
X_over,y_over=RandomOverSampler().fit_resample(X,y)
```

แบ่งข้อมูลออกเป็น 2 ชุดคือชุดทดลองและชุดทดสอบ

```
# Train Test Split
X_train,X_test,y_train,y_test = train_test_split(X_over,y_over,random_state=42,stratify=y_over)
print(f'Train shape : {X_train.shape}\nTest shape: {X_test.shape}')
```

เมื่อแบ่งเสร็จแล้วนำข้อมูลใน column “AGE” ไปทำ scaling เพื่อทำค่าให้เป็นมาตรฐาน

```
# Scaling of AGE column
scaler=StandardScaler()
X_train['AGE']=scaler.fit_transform(X_train[['AGE']])
X_test['AGE']=scaler.transform(X_test[['AGE']])
X_train.head()
```

2.3.6 สร้าง Model

นำข้อมูลชุดทดลองเข้าสู่โมเดล Logistic Regression ด้วยฟังก์ชั่นใน sklearn และนำข้อมูลชุดทดสอบเข้าสู่โมเดล Logistic Regression และทำการ predict

```
## Model Building
# Logistic Regression
param_grid={'C':[0.001,0.01,0.1,1,10,100], 'max_iter':[50,75,100,200,300,400,500,700]}
log=RandomizedSearchCV(LogisticRegression(solver='lbfgs'),param_grid,cv=5)
log.fit(X_train,y_train)
y_pred_log=log.predict(X_test)
```

หลังจากนั้นจึงนำผลลัพท์ที่ได้เข้าสู่ Confusion Matrix เพื่อประเมินประสิทธิภาพของโมเดล

```
# Model Accuracy Test
confusion_log=confusion_matrix(y_test,log.predict(X_test))
plt.figure(figsize=(8,8))
sns.heatmap(confusion_log,annot=True)
plt.xlabel("Predicted")
plt.ylabel("Actual")
plt.show()
print(classification_report(y_test,y_pred_log))
```

บทที่ 3

การประยุกต์ใช้ทฤษฎี

3.1 การประยุกต์ใช้ทฤษฎีเวกเตอร์

3.1.1 Pearson’s Similarity

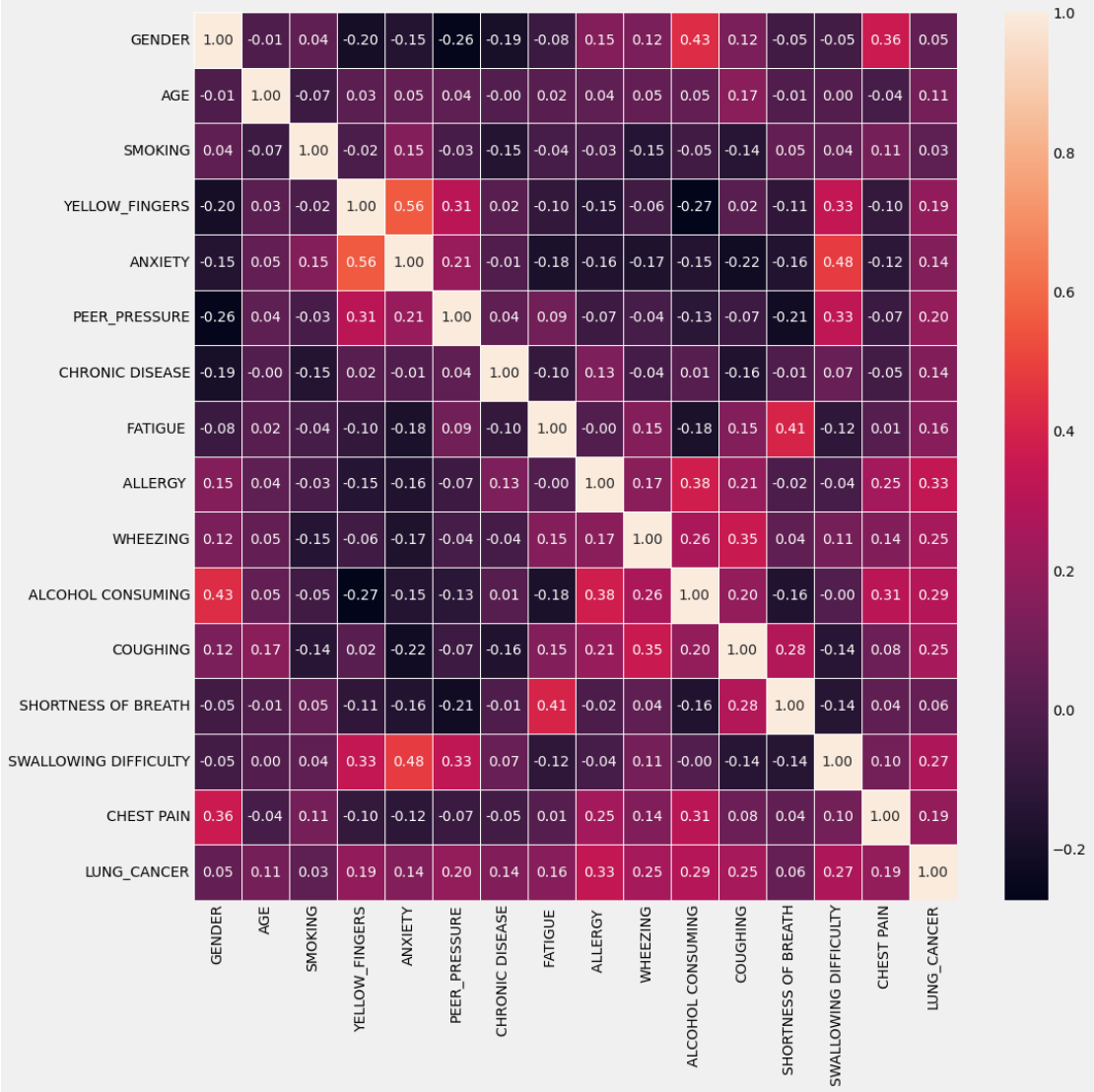
ใช้ในการหาค่าความสัมพันธ์ของข้อมูลระหว่าง 2 ตัวแปร ที่มีค่าระหว่าง -1.0 ถึง +1.0 เป็นการวิเคราะห์ความสัมพันธ์ของข้อมูลในลักษณะแบบเส้นตรงเท่านั้น

3.1.1.1 ขั้นตอนการคำนวณ

```
plt.figure(figsize=(15,15))
sns.heatmap(df.corr(),annot=True,linewidth=0.5,fmt='0.2f')
```

เป็นวิธีการเขียนโค้ดเพื่อที่จะหาค่าของ Pearson’s Similarity

โดยบรรทัดแรก คือ การกำหนดขนาดตาราง 15x15 บรรทัดที่สอง เป็นวิธีการสร้าง heatmap แล้วทำการคำนวณชุดข้อมูลด้วยฟังก์ชัน corr() เป็นการหาค่าแบบ Peason’s Similarity ที่จะมีค่าอยู่ระหว่าง -1.0 ถึง +1.0 หากค่าใกล้เคียง -1.0 ตัวแปรทั้งสองตัวแปรผกผันกัน แต่ถ้ามีค่าใกล้ +1.0 ตัวแปรทั้งสองจะแปรผันตรงกัน แล้วจะได้ผลลัพธ์ดังนี้


GENDER 1.00 -0.01 0.04 -0.20 -0.15 -0.26 -0.19 -0.08 0.15 0.12 0.43 0.12 -0.05 -0.05 0.36 0.05
AGE -0.01 1.00 -0.07 0.03 0.05 0.04 -0.00 0.02 0.04 0.05 0.05 0.17 -0.01 0.00 -0.04 0.11
SMOKING 0.04 -0.07 1.00 -0.02 0.15 -0.03 -0.15 -0.04 -0.03 -0.15 -0.05 -0.14 0.05 0.04 0.11 0.03
YELLOW_FINGERS -0.20 0.03 -0.02 1.00 0.56 0.31 0.02 -0.10 -0.15 -0.06 -0.27 0.02 -0.11 0.33 -0.10 0.19
ANXIETY -0.15 0.05 0.15 0.56 1.00 0.21 -0.01 -0.18 -0.16 -0.17 -0.15 -0.22 -0.16 0.48 -0.12 0.14
PEER_PRESSURE -0.26 0.04 -0.03 0.31 0.21 1.00 0.04 0.09 -0.07 -0.04 -0.13 -0.07 -0.21 0.33 -0.07 0.20
CHRONIC DISEASE -0.19 -0.00 -0.15 0.02 -0.01 0.04 1.00 -0.10 0.13 -0.04 0.01 -0.16 -0.01 0.07 -0.05 0.14
FATIGUE -0.08 0.02 -0.04 -0.10 -0.18 0.09 -0.10 1.00 -0.00 0.15 -0.18 0.15 0.41 -0.12 0.01 0.16
ALLERGY 0.15 0.04 -0.03 -0.15 -0.16 -0.07 0.13 -0.00 1.00 0.17 0.38 0.21 -0.02 -0.04 0.25 0.33
WHEEZING 0.12 0.05 -0.15 -0.06 -0.17 -0.04 -0.04 0.15 0.17 1.00 0.26 0.35 0.04 0.11 0.14 0.25
ALCOHOL CONSUMING 0.43 0.05 -0.05 -0.27 -0.15 -0.13 0.01 -0.18 0.38 0.26 1.00 0.20 -0.16 -0.00 0.31 0.29
COUGHING 0.12 0.17 -0.14 0.02 -0.22 -0.07 -0.16 0.15 0.21 0.35 0.20 1.00 0.28 -0.14 0.08 0.25
SHORTNESS OF BREATH -0.05 -0.01 0.05 -0.11 -0.16 -0.21 -0.01 0.41 -0.02 0.04 -0.16 0.28 1.00 -0.14 0.04 0.06
SWALLOWING DIFFICULTY -0.05 0.00 0.04 0.33 0.48 0.33 0.07 -0.12 -0.04 0.11 -0.00 -0.14 -0.14 1.00 0.10 0.27
CHEST PAIN 0.36 -0.04 0.11 -0.10 -0.12 -0.07 -0.05 0.01 0.25 0.14 0.31 0.08 0.04 0.10 1.00 0.19
LUNG_CANCER 0.05 0.11 0.03 0.19 0.14 0.20 0.14 0.16 0.33 0.25 0.29 0.25 0.06 0.27 0.19 1.00
GENDER
AGE
SMOKING
YELLOW_FINGERS
ANXIETY
PEER_PRESSURE
CHRONIC DISEASE
FATIGUE
ALLERGY
WHEEZING
ALCOHOL CONSUMING
COUGHING
SHORTNESS OF BREATH
SWALLOWING DIFFICULTY
CHEST PAIN
LUNG_CANCER
1.0
0.8
0.6
0.4
0.2
0.0
−0.2

3.2 การประยุกต์ใช้ทฤษฎีเมทริกซ์

3.2.1 Logistic Regression

ใช้ประเมินความน่าจะเป็นของผลลัพธ์ที่จะเกิดขึ้น เช่น การโหวตหรือไม่โหวต โดยชุดข้อมูลของเรา ค่าของตัวแปรต้นแต่ละตัวที่เป็นได้ คือ 1(Yes) / 0(No) โดยผลลัพธ์ที่ได้จะมีค่าระหว่าง 0 ถึง 1 เนื่องจากมันเป็นความน่าจะเป็น

3.2.1.1 ขั้นตอนการคำนวณ

```
param_grid={'C':[0.001,0.01,0.1,1,10,100],
'max_iter':[50,75,100,200,300,400,500,700]}

log=RandomizedSearchCV(LogisticRegression(solver='lbfgs'),param_grid,cv=5)
log.fit(X_train,y_train)
print(log.score(X_train,y_train)
```

ทำการ RandomizedSearchCV ใช้ ค่าประมาณ Logistic Regression และ ตัวกระจายค่า เป็น param_grid เราใส่มาเพื่อเพิ่มประสิทธิภาพในการ predict ให้แม่นยำมากยิ่งขึ้น หลังจากนั้นทำการ fit ข้อมูล X_train,y_train แล้วแสดงผลลัพธ์ข้อมูลของ Logistic Regression ได้ดังนี้

```
0.9243697478991597
```

3.2.1 Confusion Matrix

ใช้ในการประเมินผลลัพธ์ของการ Prediction ที่ทำนายจาก Model ที่เราสร้างขึ้น โดยจะแบ่งเป็นตาราง เมื่อนำไปใช้รวมกับ Model แล้วจะได้เป็นค่า Accuracy และ Precision ของ Model ที่เราทำขึ้น

3.2.1.1 ขั้นตอนการคำนวณ

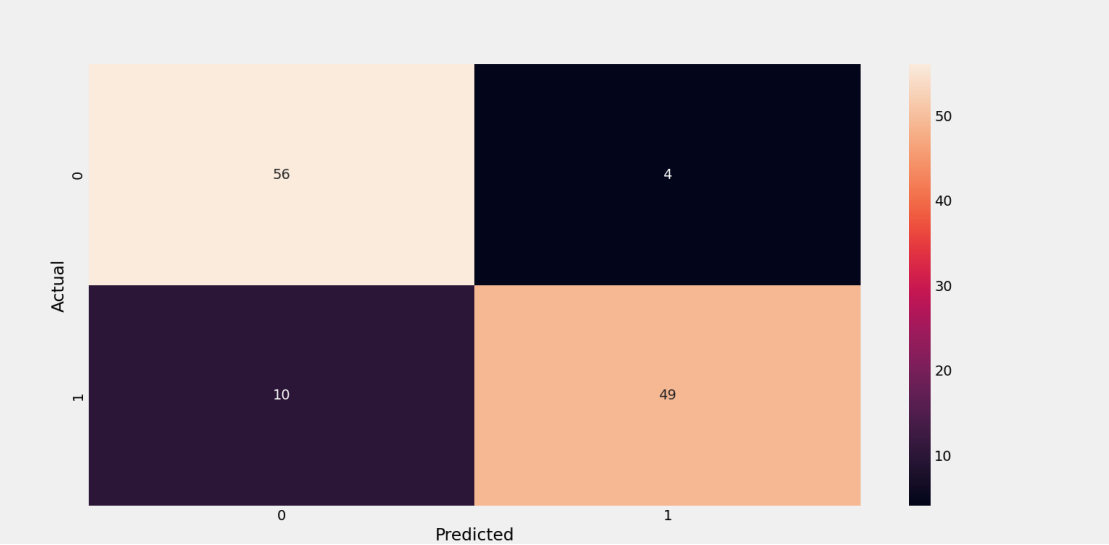
```
param_grid={'C':[0.001,0.01,0.1,1,10,100],
'max_iter':[50,75,100,200,300,400,500,700]}

log=RandomizedSearchCV(LogisticRegression(solver='lbfgs'),param_grid,cv=5)
log.fit(X_train,y_train)
log.score(X_train,y_train)
y_pred_log=log.predict(X_test)
confusion_log=confusion_matrix(y_test,log.predict(X_test))
plt.figure(figsize=(8,8))
sns.heatmap(confusion_log,annot=True)
plt.xlabel("Predicted")
plt.ylabel("Actual")

print(classification_report(y_test,y_pred_log))
```

สร้าง confusion matrix โดยใช้ฟังก์ชัน confusion matrix โดยใช้พารามิเตอร์เป็น y_test (ค่า predict lung_cencer) และ log.predict(X_test) (เป็นการ predict lung_cencer) หลังจากนั้นสร้าง modelแบบheatmap และ ทำการแสดงผลลัพธ์ของ y_test และy_pred_log(log.predict(X_test) ก็จะได้ผลลัพธ์ออกมาดังนี้

	precision	recall	f1-score	support
0	0.86	0.95	0.90	60
1	0.94	0.85	0.89	59
accuracy			0.90	119
macro avg	0.90	0.90	0.90	119
weighted avg	0.90	0.90	0.90	119



model confusion matrix

บทที่ 4

ผลการทดลอง

4.1 ผลการทดลองขั้นตอนย่อยที่ 1

จากการค้นหาข้อมูลที่จะนำมาใช้ในการทำนาย พบว่ามี Dataset จาก Kaggle ที่เกี่ยวกับโรคมะเร็งปอด โดยมีข้อมูลภายใน Dataset ดังนี้

จำนวนข้อมูลทั้งหมดใน Dataset : 309 ข้อมูล

จำนวนหัวข้อทั้งหมด : 16 หัวข้อ แบ่งออกเป็น

1. Gender: M(male), F(female)
2. Age: Age of the patient
3. Smoking: YES=2, NO=1.
4. Yellow fingers: YES=2, NO=1.
5. Anxiety: YES=2, NO=1.
6. Peer pressure: YES=2, NO=1.
7. Chronic Disease: YES=2, NO=1.
8. Fatigue: YES=2, NO=1.
9. Allergy: YES=2, NO=1.
10. Wheezing: YES=2, NO=1.
11. Alcohol: YES=2, NO=1.
12. Coughing: YES=2, NO=1.
13. Shortness of Breath: YES=2, NO=1.
14. Swallowing Difficulty: YES=2, NO=1.
15. Chest pain: YES=2, NO=1.
16. Lung Cancer: YES, NO.

4.2 ผลการทดลองขั้นตอนย่อยที่ 2

หลังจาก Import ผ่าน pandas แล้ว จึงทำการตรวจสอบข้อมูลที่นำเข้ามาดังนี้

4.2.1 ตรวจสอบ Shape ของข้อมูล

ตรวจสอบโดยใช้คำสั่ง df.shape เพื่อแสดง shape ของ Dataset

```
# Print shape of data
print(df.shape)
```

Output:

```
(309, 16)
```

ผลลัพธ์ที่ได้เป็นจำนวน row ทั้งหมด 309 row และมีจำนวน Columns ทั้งหมด 16 columns

แบ่งเป็นข้อมูลแยกแต่ละคนตาม row

4.2.2 ตรวจสอบสถิติของข้อมูลต่าง ๆ

ตรวจสอบข้อมูลทางสถิติเช่น จำนวน ค่าเฉลี่ย ค่าสูงสุด ต่ำสุด และสัดส่วนของข้อมูล

```
# Analysis numeriacal columns
print(df.describe())
```

Output:

	AGE	SMOKING	YELLOW_FINGERS	ANXIETY	...	COUGHING	SHORTNESS OF BREATH	SWALLOWING DIFFICULTY	CHEST PAIN
count	309.000000	309.000000	309.000000	309.000000	...	309.000000	309.000000	309.000000	309.000000
mean	62.673139	1.563107	1.569579	1.498382	...	1.579288	1.640777	1.469256	1.556634
std	8.210301	0.496806	0.495938	0.500808	...	0.494474	0.480551	0.499863	0.497588
min	21.000000	1.000000	1.000000	1.000000	...	1.000000	1.000000	1.000000	1.000000
25%	57.000000	1.000000	1.000000	1.000000	...	1.000000	1.000000	1.000000	1.000000
50%	62.000000	2.000000	2.000000	1.000000	...	2.000000	2.000000	1.000000	2.000000
75%	69.000000	2.000000	2.000000	2.000000	...	2.000000	2.000000	2.000000	2.000000
max	87.000000	2.000000	2.000000	2.000000	...	2.000000	2.000000	2.000000	2.000000

ผลลัพธ์ที่ได้เป็นข้อมูลทางสถิติของ Dataset สามารถเขียนออกมาเป็นตารางได้ดังนี้

	AGE	SMOKING	YELLOW FINGERS	ANXIETY	PEER PRESSURE	CHRONIC DISEASE	FATIGUE
count	309.000000	309.000000	309.000000	309.000000	309.000000	309.000000	309.000000
mean	62.673139	1.563107	1.569579	1.498382	1.501618	1.504854	1.673139
std	8.210301	0.496806	0.495938	0.500808	0.500808	0.500787	0.469827
min	21.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
25%	57.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
50%	62.000000	2.000000	2.000000	1.000000	2.000000	2.000000	2.000000
75%	69.000000	2.000000	2.000000	2.000000	2.000000	2.000000	2.000000
max	87.000000	2.000000	2.000000	2.000000	2.000000	2.000000	2.000000

	ALLERGY	WHEEZING	ALCOHOL CONSUMING	COUGHING	SHORTNESS OF BREATH	SWALLOWING DIFFICULTY	CHEST PAIN
count	309.000000	309.000000	309.000000	309.000000	309.000000	309.000000	309.000000
mean	1.556634	1.556634	1.556634	1.579288	1.640777	1.469256	1.556634
std	0.497588	0.497588	0.497588	0.494474	0.480551	0.499863	0.497588
min	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
25%	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
50%	2.000000	2.000000	2.000000	2.000000	2.000000	1.000000	2.000000
75%	2.000000	2.000000	2.000000	2.000000	2.000000	2.000000	2.000000
max	2.000000	2.000000	2.000000	2.000000	2.000000	2.000000	2.000000

4.3 ผลการทดลองขั้นตอนย่อยที่ 3

4.3.1 ตรวจสอบข้อมูลที่เป็นช่องว่างใน Dataset

```
GENDER                    0
AGE                       0
SMOKING                   0
YELLOW_FINGERS            0
ANXIETY                   0
PEER_PRESSURE             0
CHRONIC DISEASE           0
FATIGUE                   0
ALLERGY                   0
WHEEZING                  0
ALCOHOL CONSUMING         0
COUGHING                  0
SHORTNESS OF BREATH       0
SWALLOWING DIFFICULTY     0
CHEST PAIN                0
LUNG_CANCER               0
dtype: int64
```

จากตรวจสอบข้อมูลที่เป็นช่องว่าง พบว่าใน Dataset ไม่มีข้อมูลที่เป็นช่องว่าง

4.3.2 ตรวจสอบข้อมูลที่ซ้ำกันใน Dataset

```
Duplicated: 33
```

จากการตรวจสอบพบว่ามีข้อมูลซ้ำกันจำนวน 33 ข้อมูล จึงทำการ drop ส่วนที่ซ้ำกันออกไป

โดยหลังจากที่ drop ส่วนที่ซ้ำกันออกไปแล้ว เหลือข้อมูลอยู่ทั้งหมด 276 ข้อมูล

```
## Drop duplicates value
df.drop_duplicates(inplace=True)
print(df.shape)
```

```
(276, 16)
```

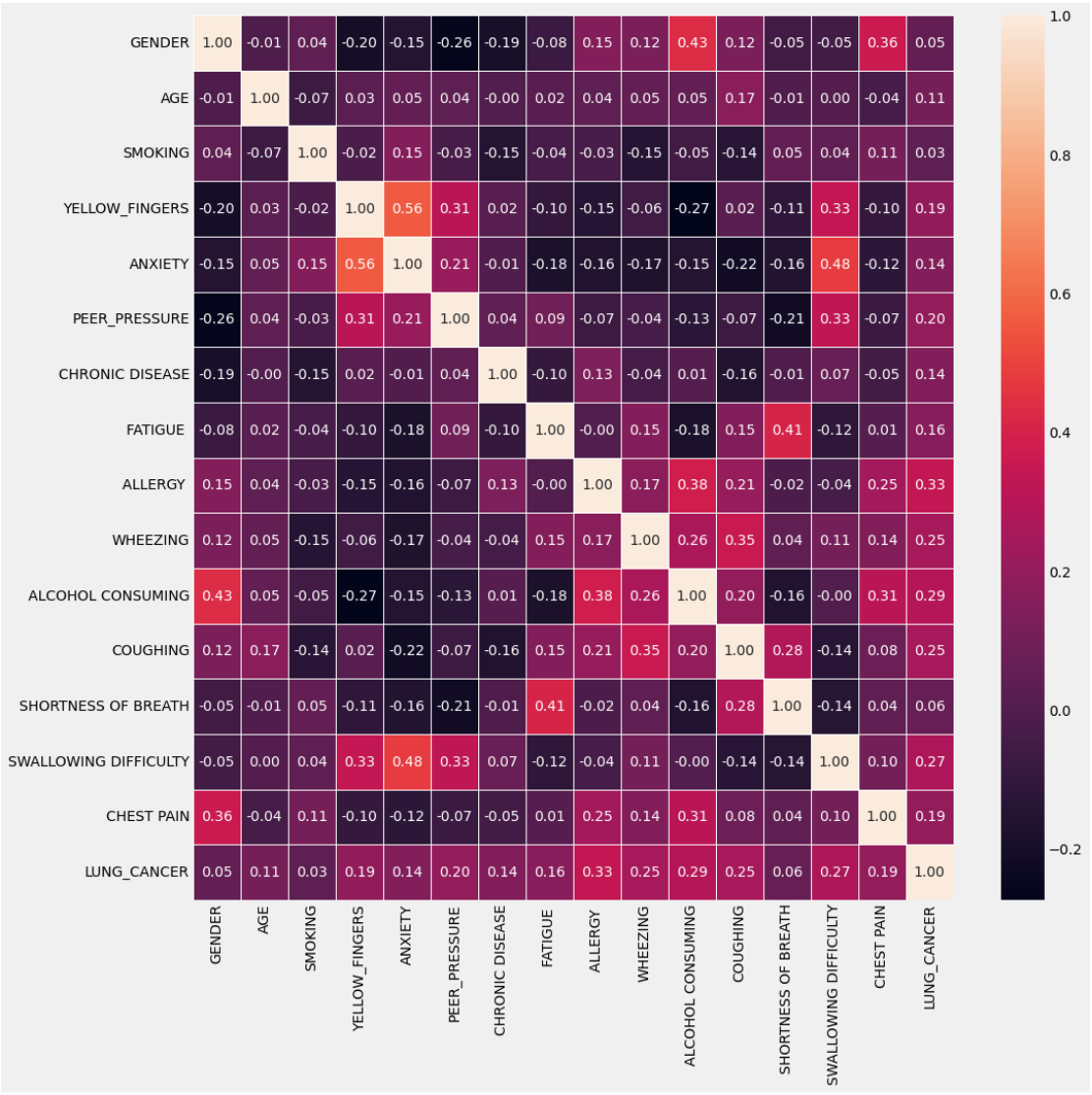
4.3.3 แปลงข้อมูลที่เป็นข้อความเป็นตัวเลข

โดยแปลงใน Columns “AGE” และ “LUNG_CANCER” หลังจากแปลงแล้ว ได้ผลลัพธ์ดังนี้

	GENDER	AGE	SMOKING	YELLOW_FINGERS	ANXIETY	...	COUGHING	SHORTNESS OF BREATH	SWALLOWING DIFFICULTY	CHEST PAIN	LUNG_CANCER
0	1	69	1	2	2	...	2	2	2	2	1
1	1	74	2	1	1	...	1	2	2	2	1
2	0	59	1	1	1	...	2	2	1	2	0
3	1	63	2	2	2	...	1	1	2	2	0
4	0	63	1	2	1	...	2	2	1	1	0

4.4 ผลการทดลองขั้นตอนย่อยที่ 4

หลังจากใช้ฟังก์ชั่นของ seaborn และ matplotlib เพื่อสร้าง Heatmap ได้ผลลัพธ์ดังนี้



ใน Heatmap เป็นหาค่าแบบ Peason’s Similarity ที่จะมีค่าอยู่ระหว่าง -1.0 ถึง +1.0 หากมีค่าใกล้เคียง -1.0 ตัวแปรทั้งสองตัวแปรผกผันกัน แต่ถ้ามีค่าใกล้ +1.0 ตัวแปรทั้งสองจะแปรผันตรงกัน

4.5 ผลการทดลองขั้นตอนย่อยที่ 5

4.5.1 เปลี่ยนค่าใน Column จาก 2,1 เป็น 1,0

หลังเปลี่ยนค่าแล้ว ได้ผลลัพธ์ดังนี้

	GENDER	AGE	SMOKING	YELLOW_FINGERS	ANXIETY	PEER_PRESSURE	...	WHEEZING	ALCOHOL CONSUMING	COUGHING	SHORTNESS OF BREATH	SWALLOWING DIFFICULTY	CHEST PAIN
0	M	69	0	1	1	0	...	1	1	1	1	1	1
1	M	74	1	0	0	0	...	0	0	0	1	1	1
2	F	59	0	0	0	1	...	1	0	1	1	0	1
3	M	63	1	1	1	0	...	0	1	0	0	1	1
4	F	63	0	1	0	0	...	1	0	1	1	0	0
5	F	75	0	1	0	0	...	1	0	1	1	0	0
6	M	52	1	0	0	0	...	1	1	1	1	0	1
7	F	51	1	1	1	1	...	0	0	0	1	1	0
8	F	68	1	0	1	0	...	0	0	0	0	0	0
9	M	53	1	1	1	1	...	0	1	0	0	1	1

โดยหากข้อมูลเป็น 0 จะหมายถึง NO และ หากข้อมูลเป็น 1 จะหมายถึง YES

4.5.2 แบ่งข้อมูลออกเป็นชุดทดลองและชุดทดสอบ

หลังจากทำการ Oversampling และแบ่งข้อมูล ได้ผลลัพธ์ดังนี้

```
Train shape : (405, 15)
Test shape: (135, 15)
```

โดยชุดทดลองจะมีข้อมูลทั้งหมด 405 ข้อมูล และชุดทดสอบจะมีข้อมูลทั้งหมด 135 ข้อมูล

4.5.3 ทำการ Scale “AGE” column

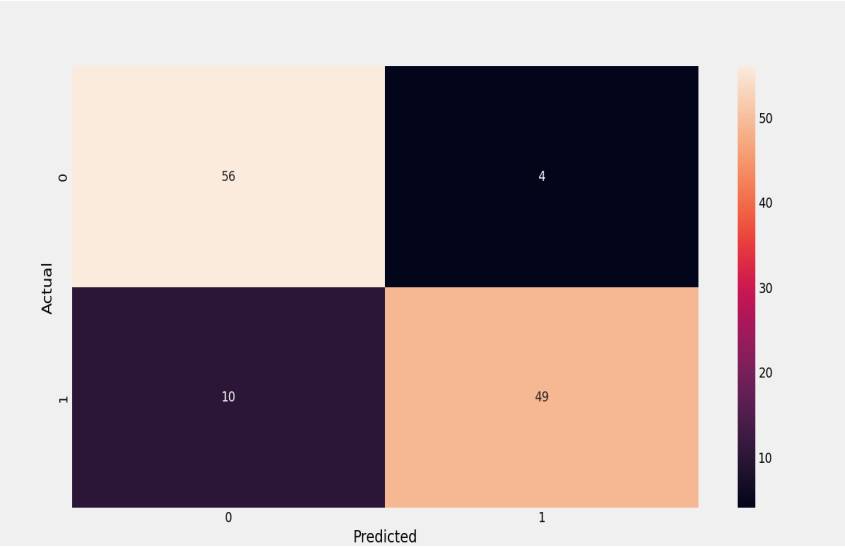
หลังจากทำการ Scale แล้วข้อมูลภายในชุดทดลองและทดสอบเป็นดังนี้

	GENDER	AGE	SMOKING	YELLOW_FINGERS	ANXIETY	...	ALCOHOL CONSUMING	COUGHING	SHORTNESS OF BREATH	SWALLOWING DIFFICULTY	CHEST PAIN
210	M	-0.964160	1	0	0	...	1	1	0	0	0
254	M	1.397405	0	1	0	...	1	1	1	0	1
83	F	2.223953	0	0	0	...	1	1	1	0	0
281	M	-0.846082	1	0	0	...	0	0	1	0	1
42	F	1.751640	0	1	1	...	1	0	0	0	0

โดยจะ Scale ข้อมูลใน column “AGE” เพื่อทำให้การทำนายมีค่าแม่นยำขึ้น

4.6 ผลการทดลองขั้นตอนย่อยที่ 6

หลังจากสร้าง Model ด้วยวิธี Logistic Regression และนำผลลัพธ์ไปเข้าสู่ Confusion Matrix แล้ว ได้ผลลัพธ์ดังนี้



	precision	recall	f1-score	support
0	0.86	0.95	0.90	60
1	0.94	0.85	0.89	59
accuracy			0.90	119
macro avg	0.90	0.90	0.90	119
weighted avg	0.90	0.90	0.90	119

โดยหลังจากทดลองแล้ว Model มีค่าจากการทดสอบทั้งหมดคือ

ค่า Accuracy (ความถูกต้องระหว่างการทำนายกับสิ่งที่เกิดขึ้นจริง) : 0.90 (90%)

ค่า Precision (ความแม่นยำ) : 0.90 (90%)

ค่า Recall (ความถูกต้องของการทำนายว่าจะเป็น “จริง” เทียบกับจำนวนครั้งของเหตุการณ์ทั้งทำนายและเกิดขึ้นว่า “เป็นจริง”) : 0.90 (90%)

ค่า F1-Score (ค่าความสามารถของ Model) : 0.90 (90%)

บทที่ 5

สรุปผลการทดลองและข้อเสนอแนะ

5.1 สรุปผลการทดลอง

จากการทำโครงงาน Lung Cancer Prediction เพื่อใช้ในการทำนายความเสี่ยงของการเกิดโรคมะเร็งปอด พบว่าเมื่อใช้ Logistic Regression เข้ามาทำนายจากข้อมูลเพศ อายุและปัจจัยความเสี่ยงต่าง ๆ ที่ส่งผลต่อการเกิดโรคมะเร็งปอด Model ที่สร้างขึ้นมีค่าความแม่นยำอยู่ที่ 90% ค่า Recall 90% โดยมีการทำนายผิดไปทั้งหมด 14 ชุดข้อมูล ซึ่งถือว่า Model ที่สร้างขึ้นนั้นมีความแม่นยำอยู่ในระดับที่ค่อนข้างแม่นยำ จึงสรุปได้ว่าการใช้ประยุกต์ใช้หลักการทางคณิตศาสตร์ Logistic Regression เพื่อนำมาสร้าง Model ในการทำนายการเกิดโรคมะเร็งปอดนั้นมีความแม่นยำมีความน่าสนใจและสามารถนำไปประยุกต์ใช้ต่อในการทำนายโรคอื่น ๆ หรือนำไปพัฒนาเพื่อทำนายข้อมูลต่าง ๆ ในชีวิตประจำวันได้

5.2 ข้อเสนอแนะ

5.2.1 ปัญหาที่พบ

1. จำนวนข้อมูลของ Dataset ระหว่างผู้ที่เป็นมะเร็งปอดกับผู้ที่ไม่เป็นมะเร็งปอดมีความจำนวนแตกต่างมากจนเกินไป

5.2.2 ข้อเสนอแนะ

1. สามารถใช้หลักการทางคณิตศาสตร์อื่น ๆ ในการทำนายได้ เช่น Support Vector Machine หรือ LGBM Classifier ซึ่งอาจจะมีค่าความแม่นยำในการทำนายมากกว่าการใช้ Logistic Regression

รายการอ้างอิง

บทความวารสาร

กาญจน์เขจร ชูชีพ. Remote Sensing Technical Note No. 5 (2018). Faculty of Forestry, Kasetsart University

สื่ออิเล็กทรอนิกส์

โรงพยาบาลศิริราช ปิยมหาราชการุณย์. มะเร็งปอด ทุกระยะดูแลได้ [ออนไลน์]. 2564, แหล่งที่มา : https://www.siphhospital.com/th/news/article/share/621/Lungcancer

Pagon Gatchalee. Confusion Matrix เครื่องมือสำคัญในการประเมินผลลัพธ์ของการทำนาย ใน Machine learning [ออนไลน์]. 2562, แหล่งที่มา : https://bit.ly/3XiNvkR

ภาคผนวก

ภาคผนวก ก

ข้อมูลโครงงาน

[1] ข้อมูลที่ใช้

https://drive.google.com/drive/folders/126eWksZUeShEqO9dJ31jPgXYrEPcu_uH?usp=sharing

[2] Source code หรือ File ที่ใช้ในการคำนวณ

https://drive.google.com/drive/folders/1y7oEsHfxLC81J2i0AGUY_dgUKCr06Q98?usp=sharing

[3] ไฟล์ประกอบอื่นๆ

https://drive.google.com/drive/folders/1_bSxvdCCfaA6UloWFg8AKa_UUDh4vs0x?usp=sharing

ภาคผนวก ข

วิดีโอและสไลด์นำเสนอโครงงาน

[1] วิดีโอนำเสนอ

https://drive.google.com/drive/folders/1UvNicYxOz9-6Z8tS1GctGfoJiGEInx1b?usp=sharing

[2] สไลด์นำเสนอ

https://drive.google.com/drive/folders/1gZHGMAz_DebRWkGM2MfIDJDZzhRDdk9p?usp=sharing