Report on the Experimental Diagnosis of Heart Disease by UCI using Machine Learning Techniques

บทคัดย่อ

จากสถิติการเสียชีวิตจากโรคหัวใจ พบว่ามีผู้เสียชีวิตกว่า 70,000 ราย คิดเป็นรายชั่วโมงคือมีผู้เสียชีวิตชั่วโมงละ 8 คน โรคหัวใจจึงเป็นปัญหาที่ต้องรัดกุมทั้งในด้านของการป้องกัน และการรักษา ดังนั้นการวินิจฉัยโรคหัวใจผ่านการใช้ Machine Learning จึงเป็นอีกหนึ่งวิธีการเพื่อนำมาทดลองในการเพิ่มความถูกต้องในการรักษาให้มากขึ้น

โดยได้มีการทดลองสร้างแบบจำลองวินิจฉัยโรคหัวใจ และเปรียบเทียบผลลัพธ์ในการหาโมเดลที่มีความแม่นยำในการวินิจฉัยโรคหัวใจ ใช้ชุดข้อมูลเกี่ยวกับค่าต่างๆ ที่เกี่ยวข้องกับการวินิจฉัยโรคหัวใจ จากการเก็บรวบรวมข้อมูลของ UCI หรือ University of California และใช้การประมวลผลข้อมูลโดยใช้โปรแกรม PyCharm CE ผ่านกระบวนการ Machine Learning แบบ Supervised Learning เลือกใช้ Classification Algorithms ทั้งหมด 3 รูปแบบ คือ Decision Tree, SVM และ Logistic Regression

ผลลัพธ์ที่ได้พบว่าทั้ง 3 Algorithms นั้นมีความสามารถในการวินิจฉัยข้อมูลที่มีความแม่นยำสูงผ่านเงื่อนไขต่างๆ และที่มีความแม่นยำสูงสุดเท่ากับ 99.03% คือโมเดลที่สร้างจาก Algorithm ของ Decision Tree และสามารถนำผลลัพธ์ดังกล่าวไปประกอบการวินิจฉัยโรคหัวใจได้อย่างมีประสิทธิภาพ

คำที่เกี่ยวข้อง

โรคหัวใจ, Machine Learning, Classification Algorithm

บทนำ

ปัญหาที่ต้องได้รับการแก้ไข คือการลดความผิดพลาดจากการตัดสินใจหรือวินิจฉัยโรคหัวใจ โดยใช้ข้อมูลและเทคโนโลยีเข้ามาช่วยเพิ่มประสิทธิภาพ การทดลองเพื่อนำความรู้ไปใช้ประโยชน์จึงเป็นสิ่งสำคัญที่จะส่งผลต่อการนำไปใช้งานจริงในอนาคต

ในส่วนของชุดข้อมูลที่นำมาใช้เป็นการเก็บรวบรวมข้อมูลของ UCI หรือ University of California และประมวลผลทางเทคโนโลยีโดยใช้ Machine Learning มาสร้างแบบจำลองการวินิจฉัยโรคหัวใจ ผ่าน Classification Algorithms ทั้ง Decision Tree, SVM และ Logistic Regression ซึ่งมีการใช้วิธีการดังกล่าวในการวิจัย และได้ผลลัพธ์ความแม่นยำที่สูงอย่างมีนัยสำคัญ (Shadman, 2018)

ผลลัพธ์จากการสร้างแบบจำลองดังกล่าวนั้นจะเป็นสิ่งที่สำคัญในด้านความรู้และสามารถนำไปศึกษาต่อในอนาคตเพื่อใช้ร่วมกับการวินิจฉัยของแพทย์อย่างแม่นยำต่อไป

วัตถุประสงค์

1. เพื่อทดลองสร้างแบบจำลองการวินิจฉัยโรคหัวใจได้อย่างมีประสิทธิภาพโดยใช้ Machine Learning
2. เพื่อเปรียบเทียบความแม่นยำในทำนายผลผ่านการสร้างแบบจำลองโดยใช้ Classification Algorithms ในหลากหลายรูปแบบ

งานวิจัยที่เกี่ยวข้อง

Harshit (2021) จากงานวิจัยได้มีการสร้างแบบจำลองการวินิจฉัยโรคหัวใจ โดยใช้ Machine Learning Algorithm ผ่าน Algorithms KNN และ Logistic Regression ได้ผลลัพธ์ออกมาในลักษณะความแม่นยำที่ดีในการวินิจฉัยโรคหัวใจจากโมเดล

Umarani (2022) จากงานวิจัยได้มีการสร้างแบบจำลองการวินิจฉัยโรคหัวใจ โดยใช้ Machine Learning Algorithms หลากหลายรูปแบบ ทั้ง SVM, Naïve Bayes และ XGBoost ผลลัพธ์โมเดลสามารถวินิจฉัยโรคได้แม่นยำอย่างมีนัยสำคัญ

Nadiah (2023) จากงานวิจัยโรคหลอดเลือดหัวใจ ที่เป็นส่วนหนึ่งของโรคหัวใจนั้น ได้มีการทดลองนำ Machine Learning Algorithm มาใช้ในการสร้างแบบจำลองเพื่อวินิจฉัย โดยได้ใช้ Algorithms ได้แก่ AdaBoost, Gradient Boost, Random Forest (RF), k-nearest neighbor (KNN), Support Vector Machine (SVM), and Decision tree เพื่อใช้ในการทดลอง ผลลัพธ์ได้แบบจำลองการวิจัยโรคหลอดเลือดหัวใจที่มีประสิทธิภาพ

Shadman (2018) จากงานวิจัยได้มีการสร้างแบบจำลองเพื่อวินิจฉัยโรคหัวใจ ผ่านการใช้ Algorithms ทั้ง Logistic Regression, SVM, Naïve Bayes และ ANN ในการสร้างแบบจำลองโมเดลขึ้นมา โดยทั้ง Logistic Regression และ SVM ได้ค่าความแม่นยำสูงมากในวินิจฉัยโรคหัวใจ

Mochammad (2022) จากงานวิจัยได้มีการสร้างแบบจำลองเพื่อวินิจฉัยโรคหัวใจ ผ่านการใช้ Algorithm คือ Logistic Regression ในการทดลอง พบว่าได้ผลลัพธ์ของโมเดลที่มีการวินิจฉัยโรคหัวใจที่มีค่าสูงอย่างมีนัยสำคัญ

Emil (2023) จากงานวิจัยได้มีการสร้างแบบจำลองเพื่อวินิจฉัยโรคหัวใจ ผ่านการใช้ Algorithm คือ Decision Tree ในการทดลองพบว่า โมเดลนั้นมีความแม่นยำสูงในการวินิจฉัยโรคหัวใจ

อธิบายชุดข้อมูล

Dataset ประกอบด้วย 14 คอลัมน์ และ 1025 แถว โดยประกอบไปด้วย

1. Target : กำหนดให้เป็นตัวแปรตาม แสดงเลข 1 หมายถึงเป็นโรคหัวใจ และเลข 0 หมายถึงไม่เป็นโรคหัวใจ
2. Age : อายุ
3. Sex : เพศ แสดงเลข 0 คือ ผู้หญิง เลข 1 คือผู้ชาย
4. Cp : รูปแบบการเจ็บหน้าอก
5. Trestbps : ค่าความดันโลหิต
6. Chol : ค่าคอเลสเตอรอล
7. Fbs : ค่าระดับน้ำตาลในเลือด หากสูงกว่า 120 mg/dl แสดงเลข 1 หากต่ำกว่าแสดงเลข 0
8. Restecg : ความผิดปกติของคลื่นไฟฟ้าหัวใจ แสดงเลข 1 กรณีผิดปกติ แสดงเลข 0 กรณีไม่ผิดปกติ
9. Thalach : ค่าการเต้นของหัวใจสูงสุด
10. Exang : ผู้ป่วยมีอาการเจ็บหน้าอกเวลาออกกำลังกาย หรือเวลาเคลื่อนไหว แสดงเลข 1 กรณีใช่ แสดงเลข 0 กรณีไม่ใช่
11. Oldpeak : ค่ากราฟคลื่นไฟฟ้าหัวใจ
12. Slope : ค่าความชันของกราฟ ณ จุดสูงสุดของคลื่นไฟฟ้าหัวใจ
13. Ca : จำนวนเส้นเลือดจากการ X-ray วินิจฉัยโรคชนิดพิเศษโดยรังสีแพทย์
14. Thal : การเกิดโลหิตจางหรือ Thalassemia

กระบวนการเตรียมข้อมูลก่อนการประมวลผล

Missing Value ได้มีการใช้โปรแกรมเพื่อ Detect Missing Value พบว่าทุกคอลัมน์ใน Dataset ไม่มี Missing Value

```
/Users/macbook/PycharmProjects/pythonProject/venv/bin/python /Users/macbook/Desktop/Final_Exam_Advanced_Python.py
age         0
sex         0
cp          0
trestbps    0
chol        0
fbs         0
restecg     0
thalach     0
exang       0
```

รูปภาพที่ 1 : แสดงผลลัพธ์ในแต่ละคอลัมน์จากการ Detect Missing Value

Outlier ได้มีการใช้โปรแกรมเพื่อ Detect Outlier พบว่ามีบางคอลัมน์ที่มีค่า Z-Score สูงกว่า 3 หรือต่ำกว่า -3 แต่จากการ Research ดูแล้วและถาม Domain Expert เพิ่มเติม ได้ความว่า ค่าดังกล่าวนั้นเป็นค่าที่พบได้ เพียงแต่อาจจะสูงหรือต่ำกว่าปกติสำหรับคนทั่วไป ไม่ใช่ค่าที่เกิดจากความผิดพลาดหรือ Error

```
outlier in dataset is [5.6, 5.6, 6.2, 6.2, 6.2, 5.6, 5.6]
```

รูปภาพที่ 2 : แสดงผลลัพธ์ของคอลัมน์ Oldpeak จากการ Detect Outlier

```
outlier in dataset is [417, 564, 409, 564, 407, 564, 407, 409, 417, 407, 407, 417, 409]
```

รูปภาพที่ 3 : แสดงผลลัพธ์ของคอลัมน์ Chol จากการ Detect Outlier

```
outlier in dataset is [192, 200, 192, 200, 192, 200, 200]
```

รูปภาพที่ 4 : แสดงผลลัพธ์ของคอลัมน์ Trestbps จากการ Detect Outlier

```
outlier in dataset is [71, 71, 71, 71]
```

รูปภาพที่ 5 : แสดงผลลัพธ์ของคอลัมน์ Thalach จากการ Detect Outlier

กระบวนการคัดเลือกตัวแปรอิสระ

ใช้วิธีการคัดเลือกตัวแปรอิสระโดยกระบวนการทำ Feature Selection ในการหาความสัมพันธ์ระหว่างตัวแปรอิสระกับตัวแปรตาม ผ่านการทำ Correlation Analysis บนโปรแกรม

โดยจากการเลือก 5 ตัวแปรอิสระที่มีความสัมพันธ์กับตัวแปรตามนั้น ได้แก่ Oldpeak, Exang, Cp, Thalacg และ Ca

แต่ในความจริงแล้ว การคัดเลือกนั้นไม่สามารถตอบได้อย่างแน่ชัดว่าควรใช้จำนวนตัวแปรอิสระเท่าไร จึงได้ให้ตัวแปรอิสระอยู่ในลักษณะของ Parameter หนึ่ง เพื่อใช้ในการทดสอบประสิทธิภาพของโมเดล

```
/Users/macbook/PycharmProjects/pythonProject/venv/bin/python /Users/macbook/Desktop/Final_Exam_Advanced_Python.py
Index(['target', 'oldpeak', 'exang', 'cp', 'thalach', 'ca'], dtype='object')

Process finished with exit code 0
```

รูปภาพที่ 6 : แสดงผลลัพธ์การเลือกจำนวนตัวแปรอิสระ 5 ตัวที่มีความสัมพันธ์กับคอลัมน์ ‘target’

กระบวนการแบ่งชุดข้อมูล

แบ่งชุดข้อมูลออกเป็น Training Set ทั้งหมด 90% และ Test Set ทั้งหมด 10% และมีการสุ่มข้อมูลต่อชุดอยู่ที่ 10 ข้อมูล เหตุผลในการแบ่งข้อมูล Training Set 90% เพื่อให้โมเดลนั้นได้เรียนรู้มากขึ้น เนื่องจากมองว่าจำนวนข้อมูลทั้งหมด ไม่ได้เพียงพอต่อการสร้างแบบจำลองที่มีประสิทธิภาพหากแบ่งเป็น Test Set มากเกินไป โดยจากผลลัพธ์มีการแบ่ง Training Set จำนวน 922 ข้อมูล และจำนวน Test Set 103 ข้อมูล

กระบวนการประมวลผลข้อมูลและทดสอบประสิทธิภาพ

จากการประมวลผลข้อมูลเพื่อสร้างโมเดล หากมีเปอร์เซ็นต์ความแม่นยำหรือมีค่า Accuracy สูง แสดงว่าโมเดลที่สร้างจาก Algorithm นั้น ทำนายผลการวินิจฉัยได้อย่างถูกต้องเกือบทั้งหมด โดยแสดงผลลัพธ์จากการหาค่า Accuracy ดังนี้

1. Logistic Regression Algorithm
   1.1 3 ตัวแปรอิสระ ได้ค่าความแม่นยำ 79.61%
   1.2 4 ตัวแปรอิสระ ได้ค่าความแม่นยำ 80.58%
   1.3 5 ตัวแปรอิสระ ได้ค่าความแม่นยำ 78.64%
2. Decision Tree Algorithm
   2.1 3 ตัวแปรอิสระ ได้ค่าความแม่นยำ 84.47%
   2.2 4 ตัวแปรอิสระ ได้ค่าความแม่นยำ 98.06%
   2.3 5 ตัวแปรอิสระ ได้ค่าความแม่นยำ 99.03%
3. SVM Algorithm
   3.1 3 ตัวแปรอิสระ ได้ค่าความแม่นยำ 79.61%
   3.2 4 ตัวแปรอิสระ ได้ค่าความแม่นยำ 65.05%
   3.3 5 ตัวแปรอิสระ ได้ค่าความแม่นยำ 66.02%

สรุปผล

จากวัตถุประสงค์ ผลลัพธ์แสดงให้เห็นว่าโมเดลมีความแม่นยำที่สูงอย่างมีนัยสำคัญในการวินิจฉัยโรคหัวใจ และผลลัพธ์ในแต่ละการสร้างแบบจำลองผ่าน Algorithm ที่แตกต่างกันพบว่า Decision Tree Algorithm มีความแม่นยำในการวินิจฉัยโรคหัวใจสูงสุดที่ 99.03%

รายงานนี้จัดทำขึ้นเพื่อให้ผู้อ่านได้รับความรู้ในการประยุกต์ใช้ Machine Learning กับการแก้ไขปัญหาโดยใช้ข้อมูล ซึ่งสามารถนำความรู้นี้ไปใช้ต่อยอดในงานวิจัยอื่นๆต่อได้ในอนาคต

อ้างอิง

Agbemade, Emil (2023). Predicting Heart Disease using Tree-based Model. Retrieved from STARS.

Anshor, Mochammad (2022). Predicting Heart Disease using Logistic Regression. Retrieved from Knowledge Engineering and Data Science

Jindal, Harshit (2021). Heart disease prediction using machine learning algorithms. Retrieved from IOP Publishing

Lapp, David (2018). Heart Disease Dataset. Retrieved from https://www.kaggle.com/datasets/johnsmith88/heart-disease-dataset/data.

Nadiah A. (2023). Advanced machine learning techniques for cardiovascular disease early detection and diagnosis. Retrieved from Journal of Big Data

Nagavelli, Umarani (2022). Machine Learning Technology-Based Heart Disease Detection Models. Retrieved from PMC

Nashif, Shadman (2018). Heart Disease Detection by Using Machine Learning Algorithms and a Real-Time Cardiovascular Health Monitoring System. Retrieved from Scientific Research Publishing

Python Code

กระบวนการเตรียมข้อมูลก่อนการประมวลผล

Missing Value

```
import pandas as pd
df = pd.read_csv("/Users/macbook/Desktop/heart.csv")
print(df.isnull().sum())
```

Outlier

```
import pandas as pd
import numpy as np
df = pd.read_csv("/Users/macbook/Desktop/heart.csv")
threshold2 = -3
threshold = 3
outlier = []
for i in df.chol:
    mean = np.mean(df.chol)
    std = np.std(df.chol)
    z = (i-mean)/std
    if z > threshold or z < threshold2:
    outlier.append(i)
print('outlier in dataset is', outlier)
```

กระบวนการคัดเลือกตัวแปรอิสระ

```
import pandas as pd
df = pd.read_csv("/Users/macbook/Desktop/heart.csv")
print(df.corr(numeric_only=True).abs().nlargest(6,'target').index)
```

กระบวนการแบ่งชุดข้อมูล

```
import pandas as pd
import numpy as np
import sklearn.model_selection as sl
df = pd.read_csv("/Users/macbook/Desktop/heart.csv")
x = pd.DataFrame(np.c_[df['exang'],df['oldpeak']],columns=['exang','oldpeak'])
y = df['target']
x_train, x_test, y_train, y_test = sl.train_test_split(x,y,test_size = 0.1,
random_state=10)
print(x_train.shape)
print(y_train.shape)
print(x_test.shape)
print(y_test.shape)
```

กระบวนการประมวลผลข้อมูลและทดสอบประสิทธิภาพ

Logistic Regression

```
import pandas as pd
import numpy as np
import sklearn.model_selection as sl
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
df = pd.read_csv("/Users/macbook/Desktop/heart.csv")
x =
pd.DataFrame(np.c_[df['exang'],df['oldpeak'],df['cp'],df['thalach']],columns=['exang','ol
dpeak','cp','thalach'])
y = df['target']
x_train, x_test, y_train, y_test = sl.train_test_split(x,y,test_size = 0.1,
random_state=10)
```

```
model = LogisticRegression().fit(x_train,y_train)
test = model.predict(x_test)
acc = accuracy_score(y_test,test)
print(acc)
```

Decision Tree

```
import pandas as pd
import numpy as np
import sklearn.model_selection as sl
from sklearn import tree
from sklearn.metrics import accuracy_score
df = pd.read_csv("/Users/macbook/Desktop/heart.csv")
x =
pd.DataFrame(np.c_[df['exang'],df['oldpeak'],df['cp'],df['thalach'],df['ca']],columns=['e
xang','oldpeak','cp','thalach','ca'])
y = df['target']
x_train, x_test, y_train, y_test = sl.train_test_split(x,y,test_size = 0.1,
random_state=10)
model = tree.DecisionTreeClassifier().fit(x_train,y_train)
test = model.predict(x_test)
acc = accuracy_score(y_test,test)
print(acc)
```

SVM

```
import pandas as pd
import numpy as np
import sklearn.model_selection as sl
from sklearn import svm
from sklearn.metrics import accuracy_score
df = pd.read_csv("/Users/macbook/Desktop/heart.csv")
```

```
x =
pd.DataFrame(np.c_[df['exang'],df['oldpeak'],df['cp']],columns=['exang','oldpeak','cp'])
y = df['target']
x_train, x_test, y_train, y_test = sl.train_test_split(x,y,test_size = 0.1,
random_state=10)
model = svm.SVC().fit(x_train,y_train)
test = model.predict(x_test)
acc = accuracy_score(y_test,test)
print(acc)
```

ตัวอย่างชุดข้อมูล

age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
52	1	0	125	212	0	1	168	0	1	2	2	3	0
53	1	0	140	203	1	0	155	1	3.1	0	0	3	0
70	1	0	145	174	0	1	125	1	2.6	0	0	3	0
61	1	0	148	203	0	1	161	0	0	2	1	3	0
62	0	0	138	294	1	1	106	0	1.9	1	3	2	0
58	0	0	100	248	0	0	122	0	1	1	0	2	1
58	1	0	114	318	0	2	140	0	4.4	0	3	1	0
55	1	0	160	289	0	0	145	1	0.8	1	1	3	0
46	1	0	120	249	0	0	144	0	0.8	2	0	3	0
54	1	0	122	286	0	0	116	1	3.2	1	2	2	0
71	0	0	112	149	0	1	125	0	1.6	1	0	2	1
43	0	0	132	341	1	0	136	1	3	1	0	3	0
34	0	1	118	210	0	1	192	0	0.7	2	0	2	1
51	1	0	140	298	0	1	122	1	4.2	1	3	3	0
52	1	0	128	204	1	1	156	1	1	1	0	0	0
34	0	1	118	210	0	1	192	0	0.7	2	0	2	1
51	0	2	140	308	0	0	142	0	1.5	2	1	2	1
54	1	0	124	266	0	0	109	1	2.2	1	1	3	0
50	0	1	120	244	0	1	162	0	1.1	2	0	2	1
58	1	2	140	211	1	0	165	0	0	2	0	2	1
60	1	2	140	185	0	0	155	0	3	1	0	2	0
67	0	0	106	223	0	1	142	0	0.3	2	2	2	1
45	1	0	104	208	0	0	148	1	3	1	0	2	1
63	0	2	135	252	0	0	172	0	0	2	0	2	1
42	0	2	120	209	0	1	173	0	0	1	0	2	1
61	0	0	145	307	0	0	146	1	1	1	0	3	0
44	1	2	130	233	0	1	179	1	0.4	2	0	2	1
58	0	1	136	319	1	0	152	0	0	2	2	2	0
56	1	2	130	256	1	0	142	1	0.6	1	1	1	0
55	0	0	180	327	0	2	117	1	3.4	1	0	2	0
44	1	0	120	169	0	1	144	1	2.8	0	0	1	0
50	0	1	120	244	0	1	162	0	1.1	2	0	2	1
57	1	0	130	131	0	1	115	1	1.2	1	1	3	0
70	1	2	160	269	0	1	112	1	2.9	1	1	3	0
50	1	2	129	196	0	1	163	0	0	2	0	2	1
46	1	2	150	231	0	1	147	0	3.6	1	0	2	0
51	1	3	125	213	0	0	125	1	1.4	2	1	2	1
59	1	0	138	271	0	0	182	0	0	2	0	2	1
64	1	0	128	263	0	1	105	1	0.2	1	1	3	1
57	1	2	128	229	0	0	150	0	0.4	1	1	3	0

รูปภาพที่ 7 : แสดงตัวอย่างชุดข้อมูล