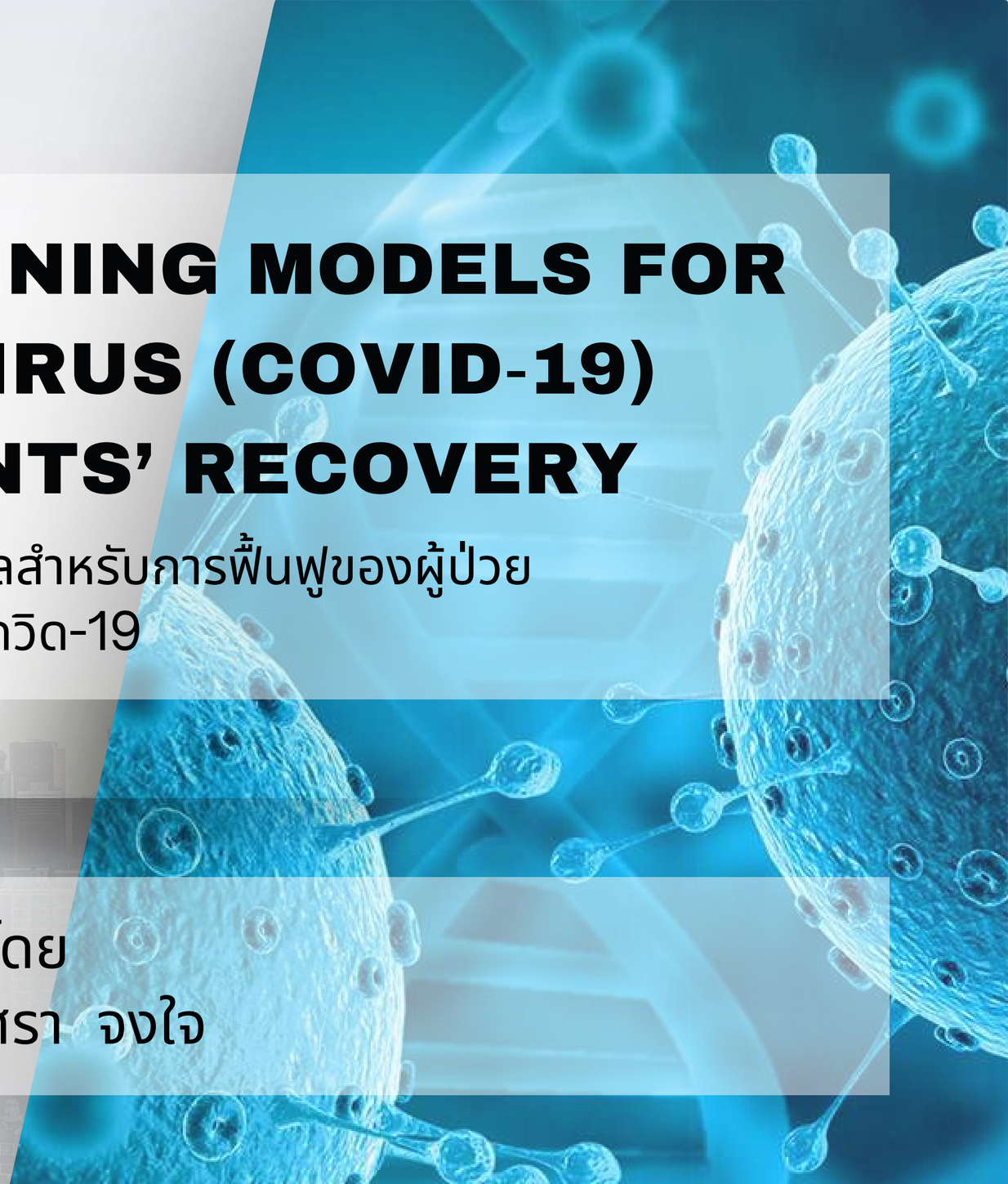
PREDICTIVE DATA MINING MODELS FOR NOVEL CORONAVIRUS (COVID-19) INFECTED PATIENTS' RECOVERY

แบบจำลองการทำเหมือนข้อมูลสำหรับการฟื้นฟูของผู้ป่วย
ที่ติดเชื้อโควิด-19



จัดทำโดย
นางสาววนิศรา จงใจ

INTRODUCTION

เนื่องด้วยในสถานการณ์โควิดในตอนนั้นมีการแพร่ระบาดอย่างรวดเร็ว บทความนี้จึงพยายามที่จะศึกษาแนวทางที่ไม่ใช่ทางคลินิก เช่น การขุดข้อมูล ปัญญาประดิษฐ์เสริม และเทคนิคปัญญาประดิษฐ์อื่นๆ เพื่อควบคุมและรับมือกับการแพร่กระจายของไวรัสโควิด-19 ที่เพิ่มมากขึ้น

บทความนี้มีวัตถุประสงค์เพื่อพัฒนาแบบจำลองการทำเหมืองข้อมูลเพื่อคาดการณ์การฟื้นตัวของผู้ป่วยที่ติดเชื้อโควิด-19 โดยใช้ชุดข้อมูลทางระบาดวิทยาจากเกาหลีใต้ อัลกอริธึมต่างๆ รวมถึง decision tree, support vector machine, naive Bayes, logistic regression, random forest, และ K-nearest neighbor.

VALUES

Decision Tree

คือ วิธีการวิเคราะห์ข้อมูลและการตัดสินใจโดยใช้โครงสร้างของต้นไม้ที่มีโหนดและกิ่ง เพื่อสร้างรูปแบบการตัดสินใจในการจำแนกข้อมูล ๆ

Naive Bayes

เป็นอัลกอริทึมในการเรียนรู้ที่ใช้ในงานจำแนกหรือจำแนกข้อมูลโดยสร้างรูปแบบการจำแนกบนข้อมูลโดยใช้หลักการของทฤษฎีการเบย์ส (Bayes' Theorem) หรือทฤษฎีความน่าจะเป็น

VALUES

Random Forest

เป็นอัลกอริทึมที่ถูกใช้ในการจำแนกข้อมูลและทำนายผลลัพธ์ โดยสร้างโมเดลจากหลายต้นไม้การตัดสินใจแล้วรวมผลลัพธ์จากต้นไม้แต่ละต้นเพื่อทำนายผลลัพธ์ที่แม่นยำมากขึ้น

KNN

เป็นอัลกอริทึมในการจำแนกและทำนายข้อมูล ที่ใช้หลักการของการเรียนรู้โดยส่งเสริมที่ความคล้ายคลึงระหว่างตัวอย่างในชุดข้อมูล

INPUT DATA ข้อมูลที่ใช้ได้มาจาก www.kaggle.com

```
Data = pd.read_csv('/content/drive/MyDrive/DPDM23_DATA/data.csv')
Data #ข้อมูลมีทั้งหมด 5165 rows × 15 columns
```

	patient_id	sex	age	country	province	city	infection_case	infected_by	contact_number	symptom_onset_date	contact_number	symptom_onset_date	confirmed_date	released_date	no-date	deceased_date	state
0	1000000001	male	50s	Korea	Seoul	Gangseo-gu	overseas inflow	NaN	75	22/1/2020	75	22/1/2020	23/1/2020	5/2/2020	13.0	NaN	released
1	1000000002	male	30s	Korea	Seoul	Jungnang-gu	overseas inflow	NaN	31	NaN	31	NaN	30/1/2020	2/3/2020	32.0	NaN	released
2	1000000003	male	50s	Korea	Seoul	Jongno-gu	contact with patient	2002000001	17	NaN	17	NaN	30/1/2020	19/2/2020	20.0	NaN	released
3	1000000004	male	20s	Korea	Seoul	Mapo-gu	overseas inflow	NaN	9	26/1/2020	9	26/1/2020	30/1/2020	15/2/2020	16.0	NaN	released
4	1000000005	female	20s	Korea	Seoul	Seongbuk-gu	contact with patient	1000000002	2	NaN	2	NaN	31/1/2020	24/2/2020	24.0	NaN	released
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
5160	7000000015	female	30s	Korea	Jeju-do	Jeju-do	overseas inflow	NaN	25	NaN	25	NaN	30/5/2020	13/6/2020	14.0	NaN	released
5161	7000000016	NaN	NaN	Korea	Jeju-do	Jeju-do	overseas inflow	NaN	NaN	NaN	NaN	NaN	16/6/2020	24/6/2020	8.0	NaN	released
5162	7000000017	NaN	NaN	Bangladesh	Jeju-do	Jeju-do	overseas inflow	NaN	72	NaN	72	NaN	18/6/2020	NaN	NaN	NaN	isolated
5163	7000000018	NaN	NaN	Bangladesh	Jeju-do	Jeju-do	overseas inflow	NaN	NaN	NaN	NaN	NaN	18/6/2020	NaN	NaN	NaN	isolated
5164	7000000019	NaN	NaN	Bangladesh	Jeju-do	Jeju-do	overseas inflow	NaN	NaN	NaN	NaN	NaN	18/6/2020	NaN	NaN	NaN	isolated

5165 rows × 15 columns

เลือกคอลัมน์ที่จะใช้

```
df = pd.DataFrame(Data)
```

```
New_Data = df[['sex', 'age', 'infection_case', 'no-date', 'state']]  #เลือกหัวข้อคอลัมน์ที่ต้องการใช้
```

```
New_Data
```

- sex = เพศ
- age = อายุ
- infection_case = ติดที่ไหน
- no-date = ระยะเวลาที่ติด
- state = สถานะ

sex	age	infection_case	no-date	state
male	50s	overseas inflow	13.0	released
male	30s	overseas inflow	32.0	released
male	50s	contact with patient	20.0	released
male	20s	overseas inflow	16.0	released
female	20s	contact with patient	24.0	released
...	...	...	...	...
female	30s	overseas inflow	14.0	released
NaN	NaN	overseas inflow	8.0	released
NaN	NaN	overseas inflow	NaN	isolated
NaN	NaN	overseas inflow	NaN	isolated
NaN	NaN	overseas inflow	NaN	isolated

เช็ค **MISSING VALUE**

```
New_Data.isnull().any() #ทุกคอลัมน์มีค่า Missing
```

```
sex                 True
age                 True
infection_case      True
no-date             True
state               True
dtype: bool
```

กำจัดค่า **MISSINGG**

แทนที่ missing ด้วยค่าที่เหมาะสม

```
New_Data = New_Data.fillna({'sex':'unknown','age':'unknown','infection_case':'unknown','no-date':0})
New_Data
```

	sex	age	infection_case	no-date	state
0	male	50s	overseas inflow	13.0	released
1	male	30s	overseas inflow	32.0	released
2	male	50s	contact with patient	20.0	released
3	male	20s	overseas inflow	16.0	released
4	female	20s	contact with patient	24.0	released
...	...	...	...	...	...
5160	female	30s	overseas inflow	14.0	released
5161	unknown	unknown	overseas inflow	8.0	released
5162	unknown	unknown	overseas inflow	0.0	isolated
5163	unknown	unknown	overseas inflow	0.0	isolated
5164	unknown	unknown	overseas inflow	0.0	isolated

5165 rows × 5 columns

เช็ค **MISSING VALUE**

```
New_Data.isnull().any() #เช็คค่า Missing อีกรอบ  #ในคอลลัมน์ state ยังมี Missing อยู่
```

```
sex                False
age                False
infection_case     False
no-date            False
state               True
dtype: bool
```

```
New_Data[New_Data['state'].isnull()] #เช็คว่าในคอลัมน์ state แถวไหนที่มีค่า Missing
```

	sex	age	infection_case	no-date	state
4045	female	20s	unknown	34.0	NaN

กำจัดค่า **MISSINGG**
โดยการ dropna

```
New_Data = New_Data.dropna()  #ทำการ dropna เพื่อที่จะตัดแถวที่มีค่า missing ออกไป
New_Data
```

	sex	age	infection_case	no-date	state
0	male	50s	overseas inflow	13.0	released
1	male	30s	overseas inflow	32.0	released
2	male	50s	contact with patient	20.0	released
3	male	20s	overseas inflow	16.0	released
4	female	20s	contact with patient	24.0	released
...	...	...	...	...	...
5160	female	30s	overseas inflow	14.0	released
5161	unknown	unknown	overseas inflow	8.0	released
5162	unknown	unknown	overseas inflow	0.0	isolated
5163	unknown	unknown	overseas inflow	0.0	isolated
5164	unknown	unknown	overseas inflow	0.0	isolated

5164 rows × 5 columns

เช็ค **MISSING VALUE**

```
New_Data.isnull().any() #เช็คค่า Missing อีกรอบ  #พบว่าไม่มีค่า Missing แล้ว
```

```
sex                 False
age                 False
infection_case      False
no-date             False
state               False
dtype: bool
```

REPLACE แทนค่าเป็นตัวเลข

```
set(New_Data['sex'])  #เช็คว่าในคอลัมน์ sex มีกลุ่มไหนบ้าง
```

```
{'female', 'male', 'unknown'}
```

```
New_Data = New_Data.replace({'female':0,'male':1,'unknown':99})
New_Data
```

- เพศหญิง = 0
- เพศชาย = 1
- unknown = 99

sex	age	infection_case	no-date	state
1	50s	overseas inflow	13.0	released
1	30s	overseas inflow	32.0	released
1	50s	contact with patient	20.0	released
1	20s	overseas inflow	16.0	released
0	20s	contact with patient	24.0	released
...	...	...	...	...
0	30s	overseas inflow	14.0	released
99	99	overseas inflow	8.0	released
99	99	overseas inflow	0.0	isolated
99	99	overseas inflow	0.0	isolated
99	99	overseas inflow	0.0	isolated

REPLACE แทนค่าเป็นตัวเลข

```
set(New_Data['age']) #เช็คว่าในคอลัมน์ age มีกลุ่มไหนบ้าง # มี 12 กลุ่ม
```

```
{'0s',
 '100s',
 '10s',
 '20s',
 '30s',
 '40s',
 '50s',
 '60s',
 '70s',
 '80s',
 '90s',
 99}
```

```
New_Data = New_Data.replace({'0s':0,'10s':1,'20s':2,'30s':3,'40s':4,'50s':5,'60s':6,'70s':7,'80s':8,'90s':9,'100s':10})
New_Data
```

sex	age	infection_case	no-date	state
1	5	overseas inflow	13.0	released
1	3	overseas inflow	32.0	released
1	5	contact with patient	20.0	released
1	2	overseas inflow	16.0	released
0	2	contact with patient	24.0	released
...	...	...	...	...
0	3	overseas inflow	14.0	released
99	99	overseas inflow	8.0	released
99	99	overseas inflow	0.0	isolated
99	99	overseas inflow	0.0	isolated
99	99	overseas inflow	0.0	isolated

REPLACE แทนค่าเป็นตัวเลข

```
set(New_Data['infection_case']) #เช็คว่าในคอลัมน์ infection_case มีกลุ่มไหนบ้าง # มี 51 กลุ่ม #ไม่ใช้คอลัมน์นี้
```

```
{99,
 'Anyang Gunpo Pastors Group',
 'Biblical Language study meeting',
 'Bonghwa Pureun Nursing Home',
 'Changnyeong Coin Karaoke',
 'Cheongdo Daenam Hospital',
 'Coupang Logistics Center',
 'Daejeon door-to-door sales',
 'Daezayeon Korea',
 'Day Care Center',
 'Dongan Church',
 'Dunsan Electronics Town',
 "Eunpyeong St. Mary's Hospital",
 'Gangnam Dongin Church',
 'Gangnam Yeoksam-dong gathering',
 'Geochang Church',
 'Geumcheon-gu rice milling machine manufacture',
 'Guri Collective Infection',
 'Guro-gu Call Center',
 'Gyeongsan Cham Joeun Community Center',
 'Gyeongsan Jeil Silver Town',
 'Gyeongsan Seorin Nursing Home',
 'Itaewon Clubs',
 'KB Life Insurance',
 'Korea Campus Crusade of Christ',
 'Milal Shelter',
 'Ministry of Oceans and Fisheries',
 'Onchun Church',
 'Orange Life',
 'Orange Town',
 'Pilgrimage to Israel',
 'Richway',
 'River of Grace Community Church',
 'SMR Newly Planted Churches Group',
 'Samsung Fire & Marine Insurance',
 'Samsung Medical Center',
 'Seocho Family',
 'Seongdong-gu APT',
 'Seoul City Hall Station safety worker',
 'Shincheonji Church',
 'Suyeong-gu Kindergarten',
 'Uiwang Logistics Center',
 'Wangsung Church',
 'Yangcheon Table Tennis Club',
 'Yeonana News Class',
 'Yeongdeungpo Learning Institute',
 'Yongin Brothers',
 'contact with patient',
 'etc',
 'gym facility in Cheonan',
 'gym facility in Sejong',
 'overseas inflow'}
```

REPLACE แทนค่าเป็นตัวเลข

```
set(New_Data['state'])  #เช็คว่าในคอลัมน์ state มีกลุ่มไหนบ้าง # มี 3 กลุ่ม
```

```
{'deceased', 'isolated', 'released'}
```

```
New_Data = New_Data.replace({'deceased':0,'isolated':1,'released':2})
New_Data
```

- deceased = ตาย
- isolated = กักตัว
- released = ไม่มีความเสี่ยง

sex	age	infection_case	no-date	state
1	5	overseas inflow	13.0	2
1	3	overseas inflow	32.0	2
1	5	contact with patient	20.0	2
1	2	overseas inflow	16.0	2
0	2	contact with patient	24.0	2
...	...	...	...	...
0	3	overseas inflow	14.0	2
99	99	overseas inflow	8.0	2
99	99	overseas inflow	0.0	1
99	99	overseas inflow	0.0	1
99	99	overseas inflow	0.0	1

เลือกคอลัมน์ที่จะใช้ใหม่

```
New_Data = New_Data[['sex', 'age', 'no-date', 'state']]
New_Data
```

	sex	age	no-date	state
0	1	5	13.0	2
1	1	3	32.0	2
2	1	5	20.0	2
3	1	2	16.0	2
4	0	2	24.0	2
...	...	...	...	...
5160	0	3	14.0	2
5161	99	99	8.0	2
5162	99	99	0.0	1
5163	99	99	0.0	1
5164	99	99	0.0	1

SET NEW DATA TO SPLIT TRAIN AND TEST

```
np.random.seed(seed = 789)
```

```
A = np.random.randint(2, size = len(New_Data))
A
```

```
array([1, 0, 0, ..., 1, 0, 1])
```

```
New_train = New_Data [A==1]
New_train.shape
```

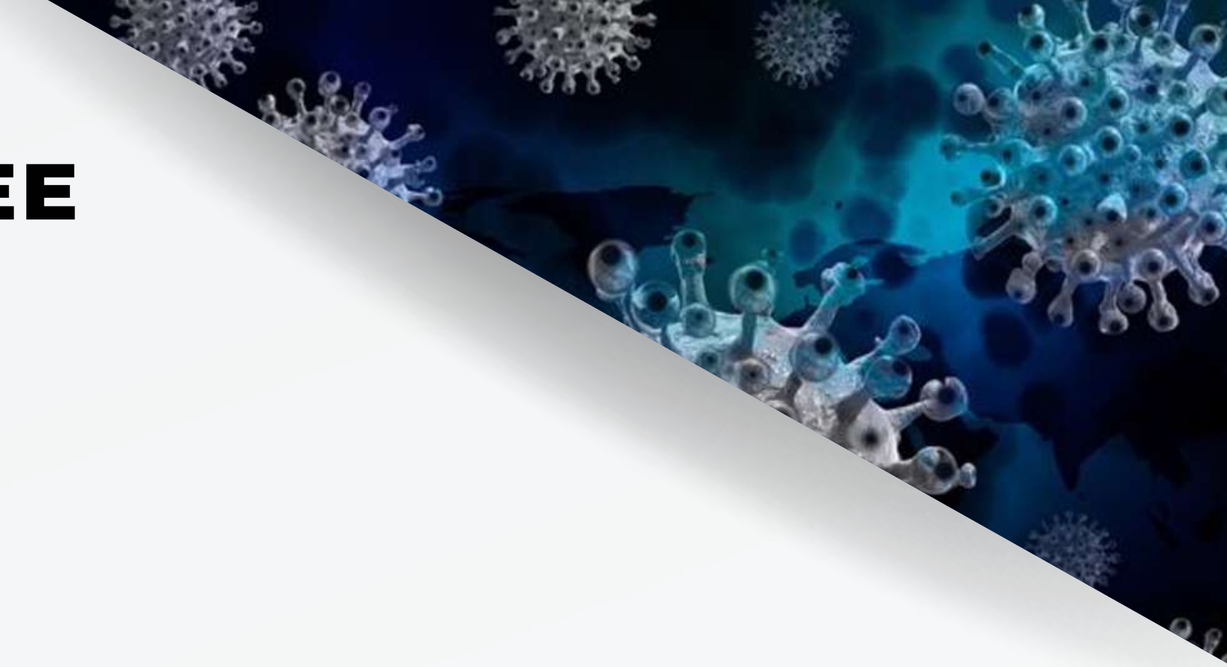
```
(2596, 4)
```

```
New_test = New_Data [A==0]
New_test.shape
```

```
(2568, 4)
```

SET DATA

```
NewX_train = New_train.iloc[:,:-1]
Newy_train = New_train.iloc[:,-1]
NewX_test = New_test.iloc[:,:-1]
Newy_test = New_test.iloc[:,-1]
```



โมเดลที่ 1 ของ DECISION TREE

DEFINE

```
from sklearn.tree import DecisionTreeClassifier
```

```
from sklearn import tree
```

```
D_tree = DecisionTreeClassifier()
```

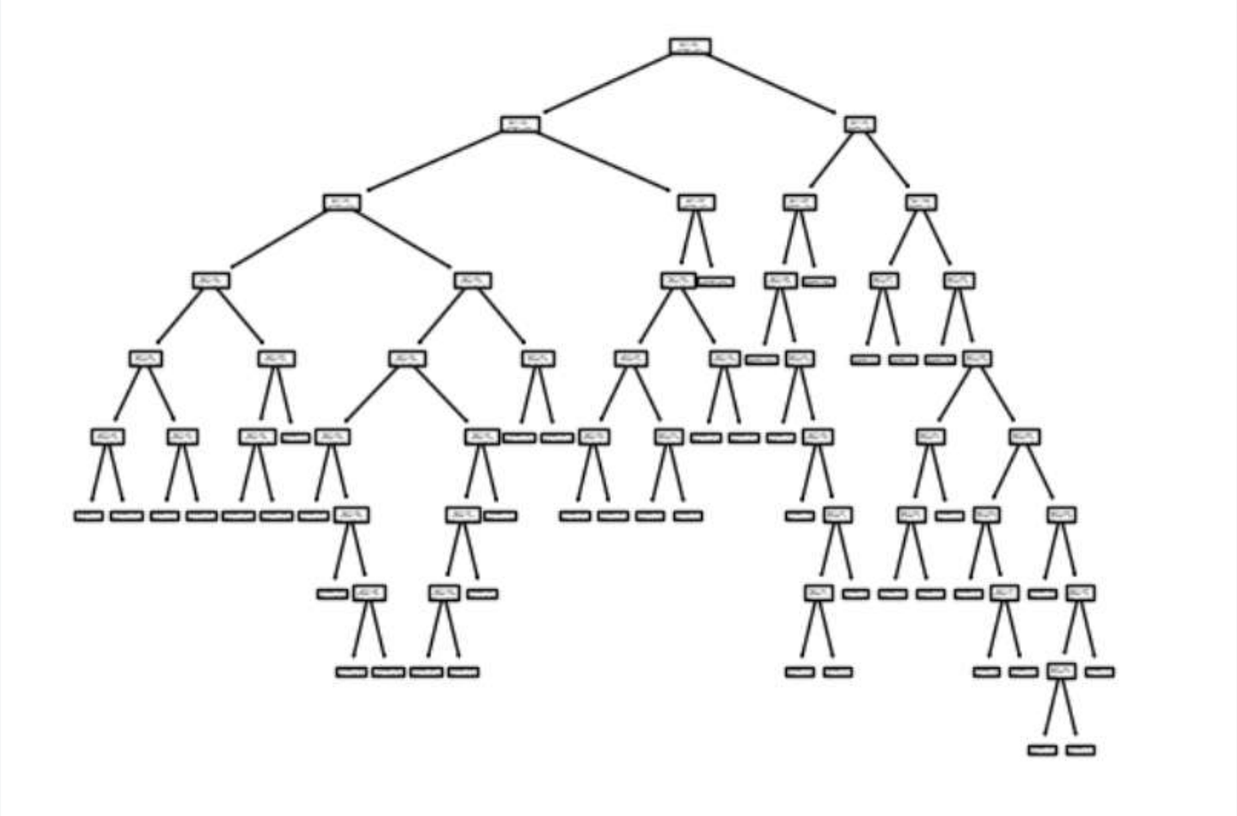
TRAIN

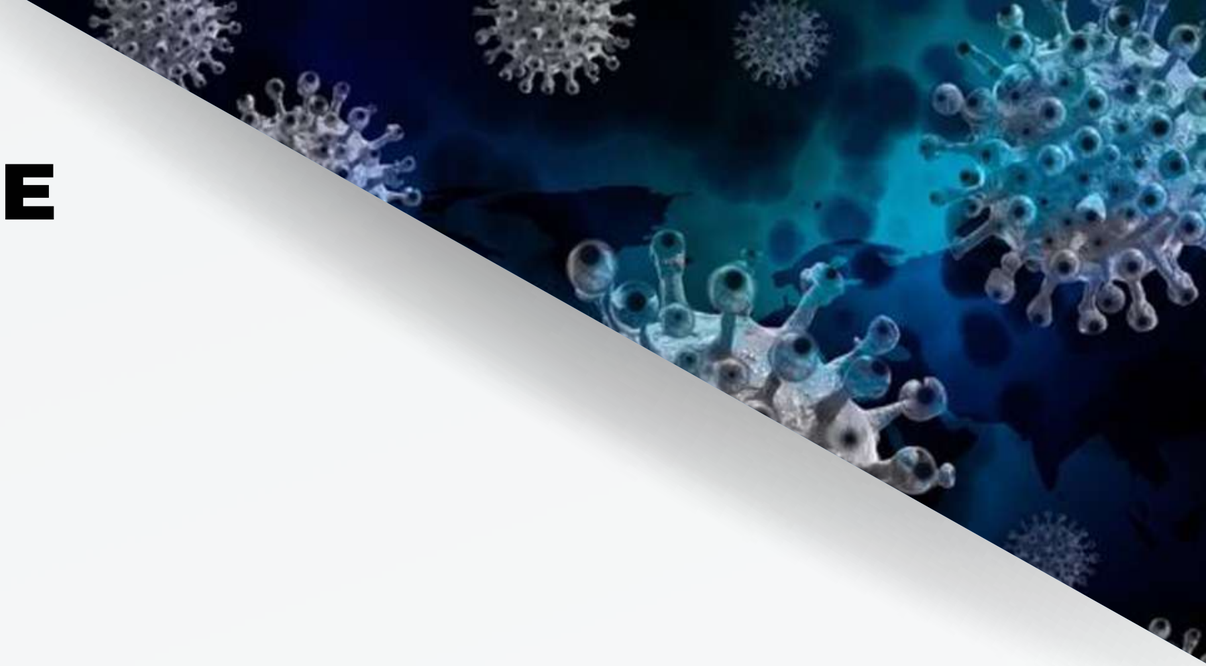
```
D_tree.fit(NewX_train,Newy_train)
```

```
DecisionTreeClassifier()
```

โมเดลที่ **1** ของ **DECISION TREE**

```
tree.plot_tree(D_tree);
```





โมเดลที่ **1** ของ **DECISION TREE**

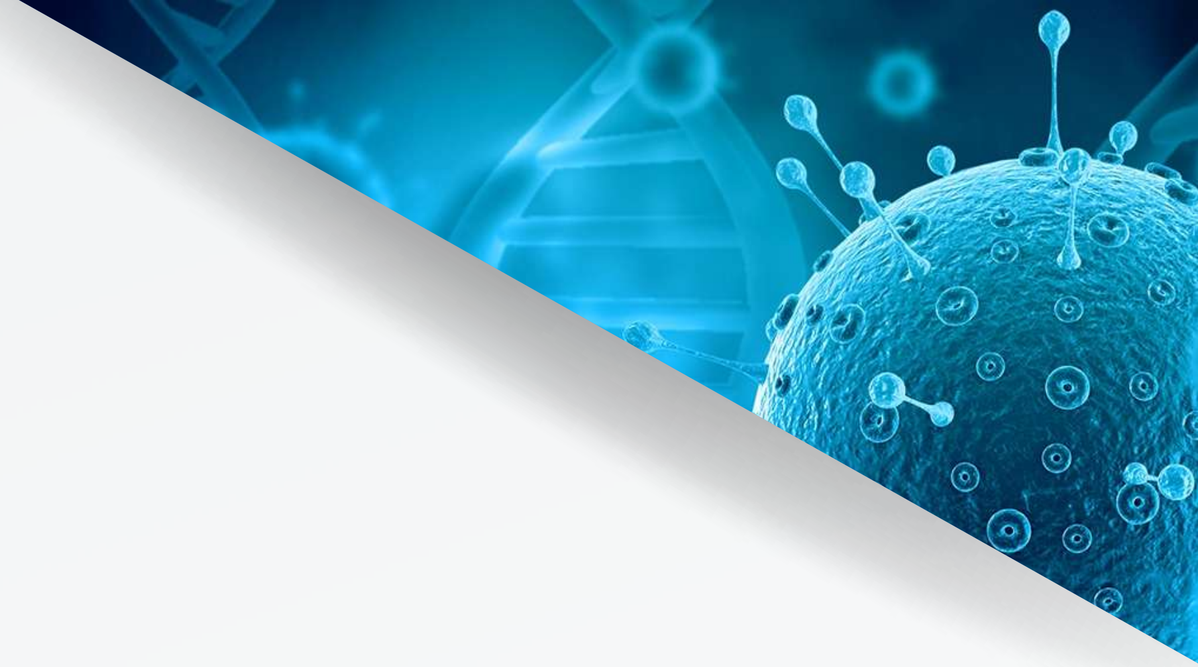
TEST

```
from sklearn.metrics import accuracy_score
```

```
y_predict1 = D_tree.predict(NewX_test)
```

```
accuracy_score(Newy_test, y_predict1) #ค่าความแม่นยำ
```

```
0.7133956386292835
```



โมเดลที่ 2 ของ NAIVE BAYES

IMPORT

```
from sklearn.naive_bayes import GaussianNB
```

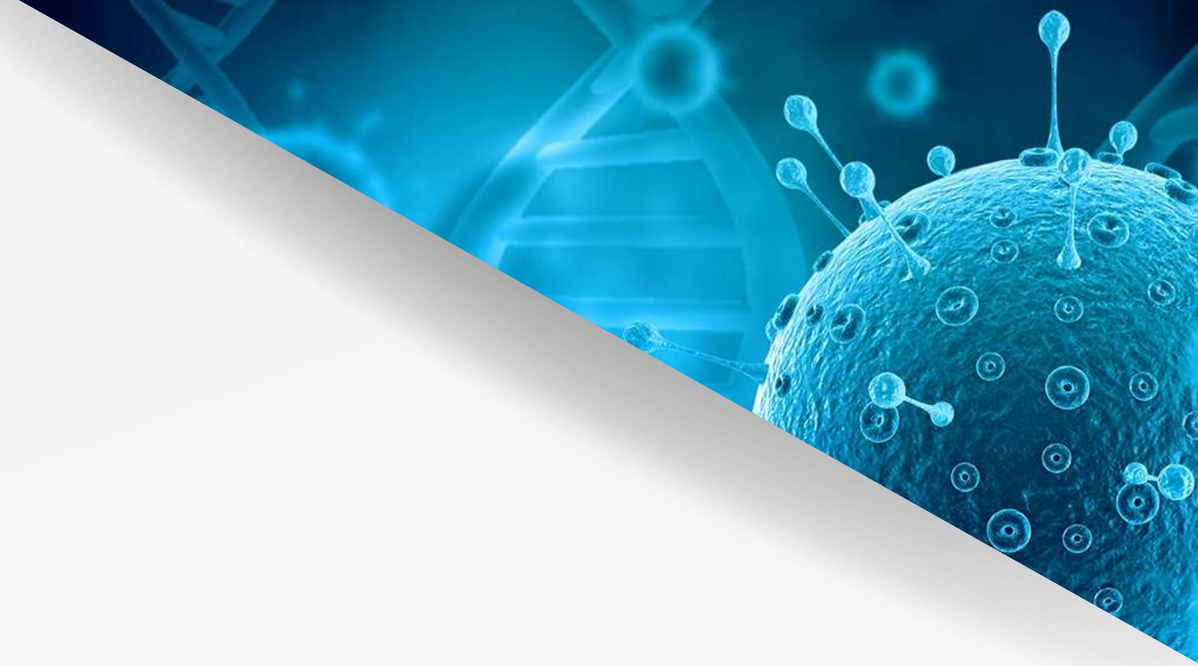
DEFINE

```
N_Bayes = GaussianNB()
```

TRAIN

```
N_Bayes.fit(NewX_train, Newy_train)
```

```
GaussianNB()
```



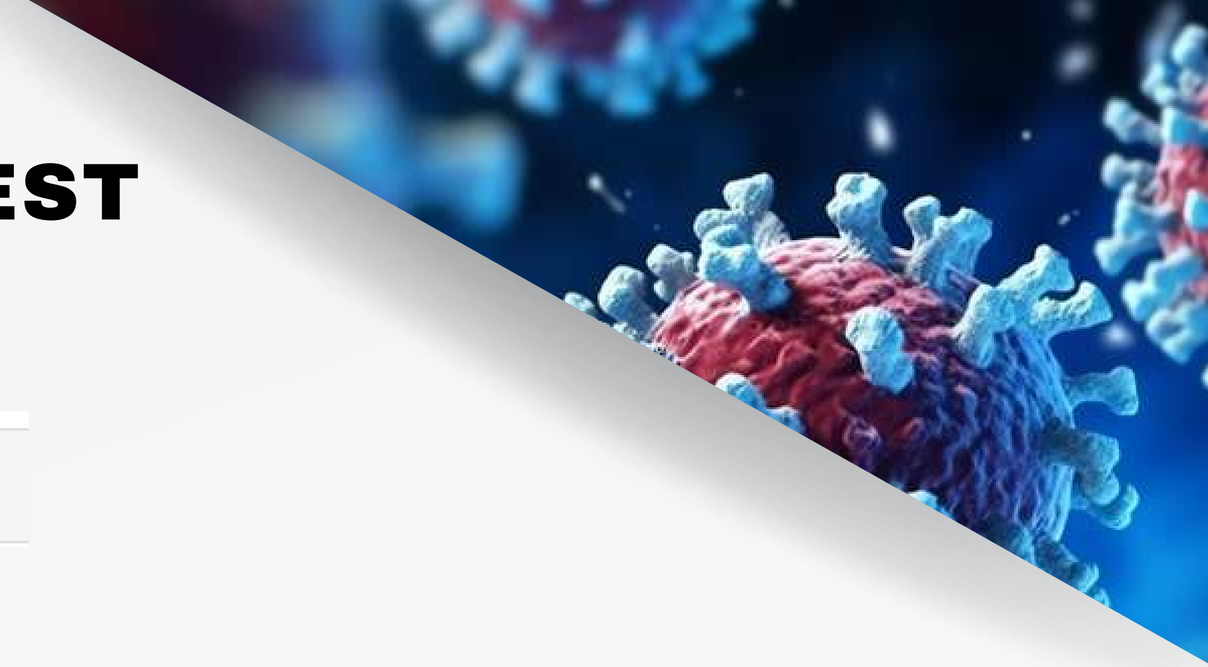
โมเดลที่ 2 ของ NAIVE BAYES

TEST

```
y_predict2 = N_Bayes.predict(NewX_test)
```

```
from sklearn.metrics import accuracy_score
accuracy_score(Newy_test, y_predict2)
```

```
0.721183800623053
```



โมเดลที่ 3 ของ RANDOM FOREST

IMPORT

```
from sklearn.ensemble import RandomForestClassifier
```

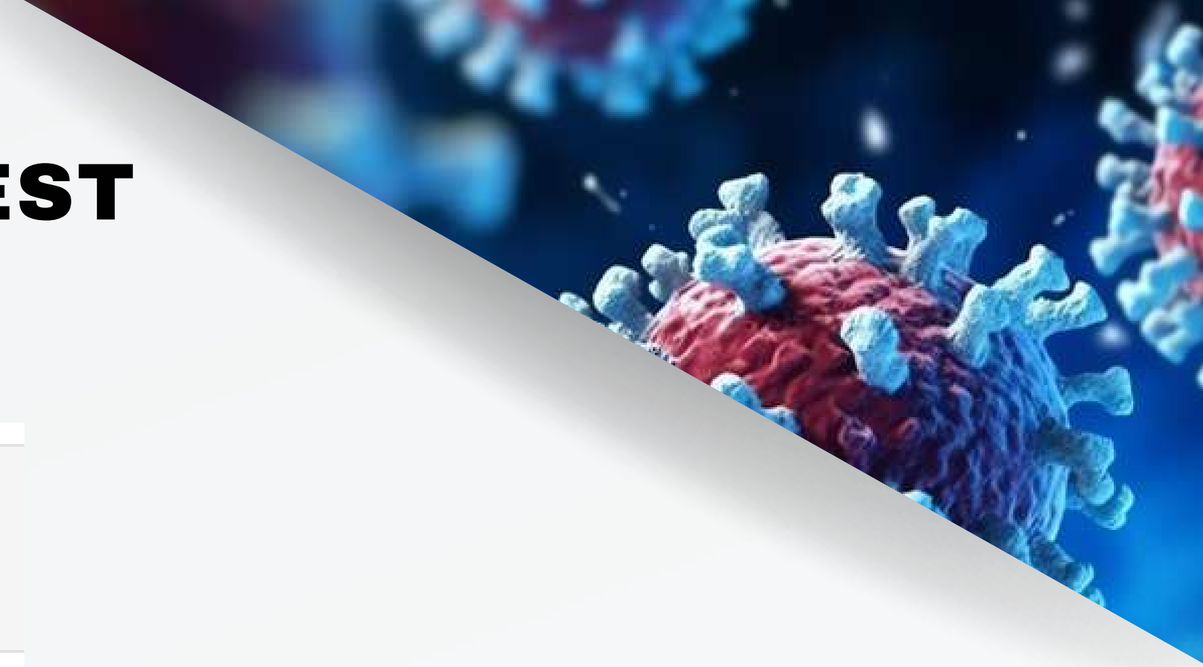
DEFINE

```
R_Forest = RandomForestClassifier()
```

TRAIN

```
R_Forest.fit(NewX_train, Newy_train)
```

```
RandomForestClassifier()
```

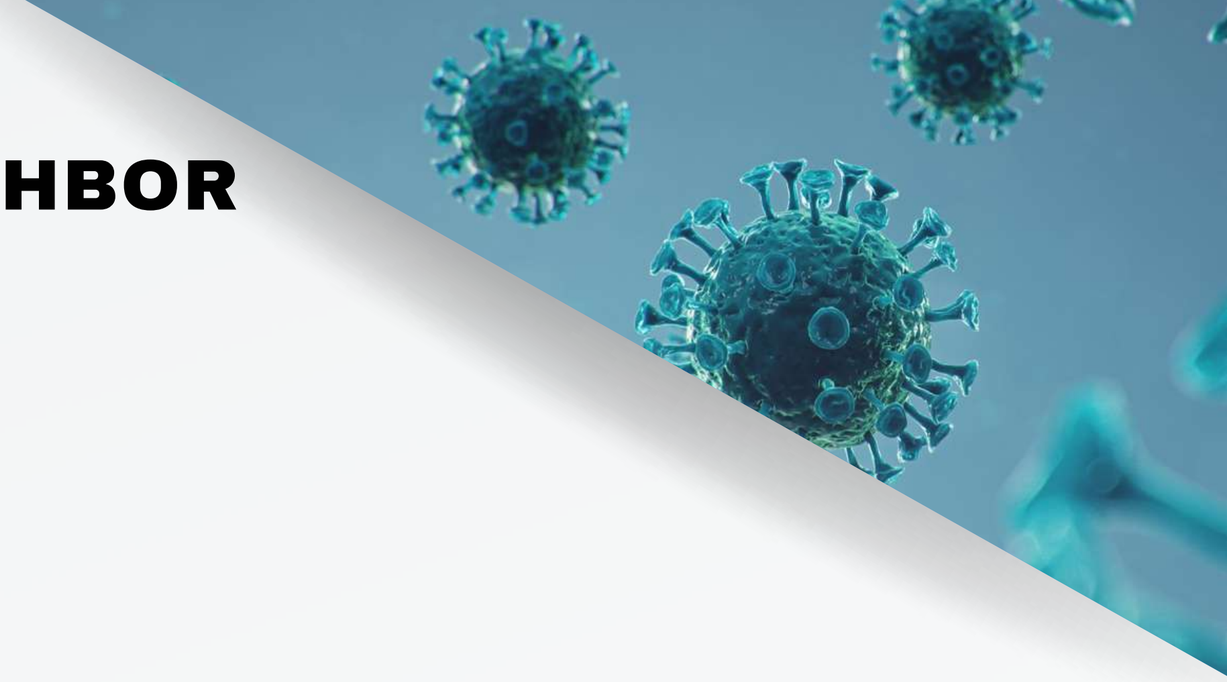


โมเดลที่ **3** ของ **RANDOM FOREST**

TEST

```
Y_predict3 = R_Forest.predict(NewX_test)
accuracy_score(Newy_test,Y_predict3)
```

```
0.7184579439252337
```



โมเดลที่ 4 ของ K-NEAREST NEIGHBOR

IMPORT

```
from sklearn.neighbors import KNeighborsClassifier
```

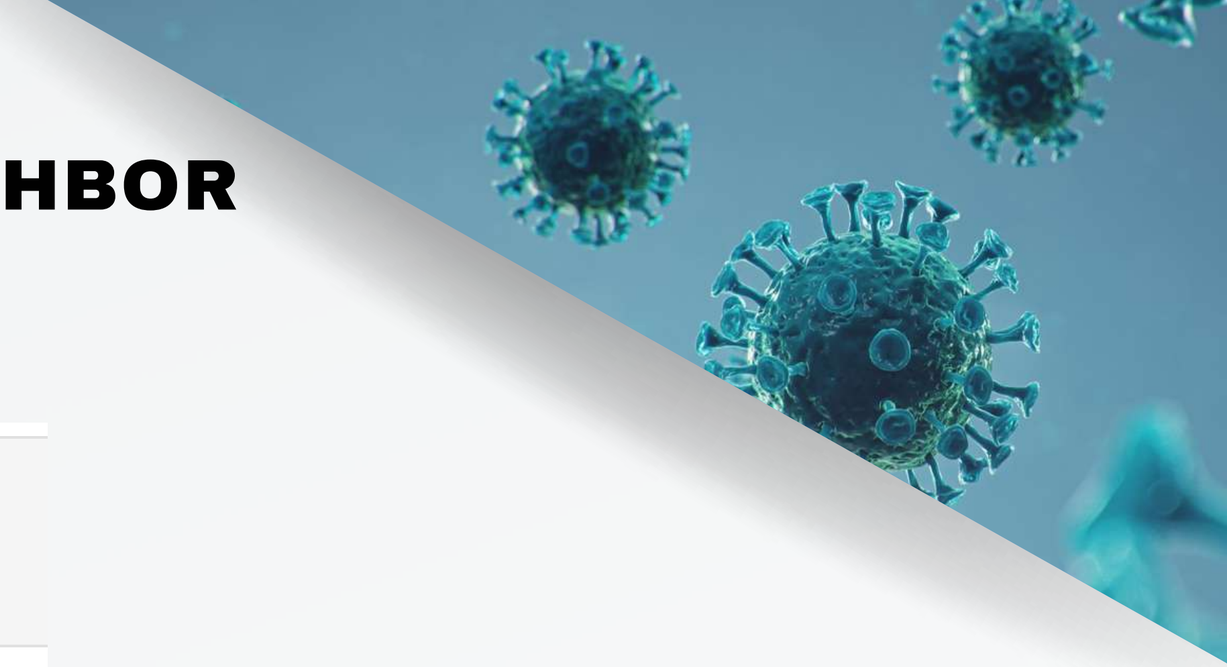
DEFINE

```
KNN = KNeighborsClassifier(n_neighbors=3)
```

TRAIN

```
KNN.fit(NewX_train, Newy_train)
```

```
KNeighborsClassifier(n_neighbors=3)
```



โมเดลที่ **4** ของ **K-NEAREST NEIGHBOR**

TEST

```
Y_predict4 = KNN.predict(NewX_test)
accuracy_score(Newy_test,Y_predict4)
```

```
0.6203271028037384
```

VALIDATE MODEL

โมเดลที่ **1** ของ **Decision Tree**

```
y_predict1 = D_tree.predict(NewX_test)
accuracy_score(Newy_test, y_predict1)
```

```
0.7133956386292835
```

โมเดลที่ **2** ของ **Naive Bayes**

```
y_predict2 = N_Bayes.predict(NewX_test)
accuracy_score(Newy_test, y_predict2)
```

```
0.721183800623053
```

โมเดลที่ **3** ของ **Random Forest**

```
Y_predict3 = R_Forest.predict(NewX_test)
accuracy_score(Newy_test,Y_predict3)
```

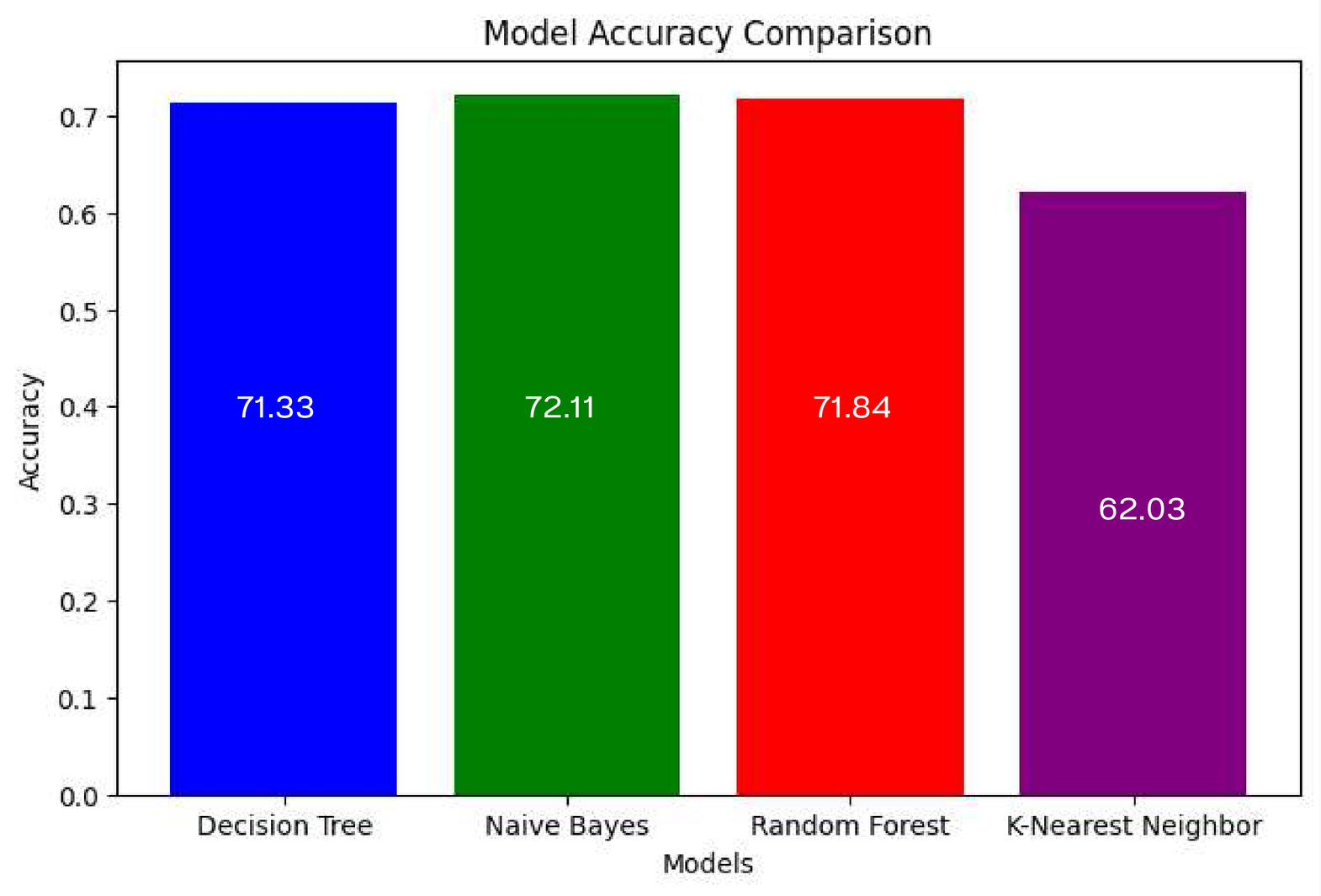
```
0.7184579439252337
```

โมเดลที่ **4** ของ **KNN**

```
Y_predict4 = KNN.predict(NewX_test)
accuracy_score(Newy_test,Y_predict4)
```

```
0.6203271028037384
```

VALIDATE MODEL



EVALUATION โมเดลที่ 2 ของ NAIVE BAYES

```
cm = confusion_matrix(Newy_test,y_predict2)
cm
```

```
array([[   0,   30,    0],
       [   0, 1080,    2],
       [   0,  684,  772]])
```

CLASSIFICATION REPORT

```
print(classification_report(Newy_test,y_predict2))
```

	precision	recall	f1-score	support
0	0.00	0.00	0.00	30
1	0.60	1.00	0.75	1082
2	1.00	0.53	0.69	1456
accuracy			0.72	2568
macro avg	0.53	0.51	0.48	2568
weighted avg	0.82	0.72	0.71	2568


THANK YOU