**Putting it All Together: Building a Data Mining System, From Data Cleaning to Model Evaluation**

Rafe Mohammed Abdul

Masters in Information Technology, Arizona State University

IFT 511 : Analyzing Big Data

Prof: Asmaa Elbadrawy

April 30, 2023

* The dataset contains 22 columns and 70692 rows.
* Below is the explanation for each row.

1. Diabetes\_binary – It is represented using a 0 or 1 where 0 refers to no diabetes while 1 refers to diabetic.
2. HighBP - It is represented using a 0 or 1 where 0 refers to low BP while 1 refers to High Bp.
3. HighChol - It is represented using a 0 or 1 where 0 refers to low Cholestrol while 1 refers to High cholestrol.
4. CholCheck - It is represented using a 0 or 1 where 0 indicates to no cholesterol check(5 years) whereas 1 is equal to yes cholesterol check(5 years).
5. BMI – Refers to body mass index.
6. Smoker - It is represented using a 0 or 1 where 0 refers to person where he didn’t smoke 100 cigarettes in his entire life while 1 refers to yes.
7. Stroke - It is represented using a 0 or 1 where 0 refers to you didn’t receive a stroke while 1 refers to yes.
8. HeartDiseasorAttack - It is represented using a 0 or 1; coronary heart disease (CHD) or myocardial infarction (MI) 0 indicates no whereas 1 represents yes.
9. PhysActivity - It is represented using a 0 or 1; physical activity in past 30 days - not including job 0 indicates no whereas 1 indicated yes
10. Fruits - It is represented using a 0 or 1; Consume Fruit one or more than 1 times per day 0 indicates no whereas 1 indicates yes
11. Veggies - It is represented using a 0 or 1; Consume Vegetables one or more than 1 times per day 0 indicates no whereas 1 indicates yes
12. HvyAlcoholConsump - It is represented using a 0 or 1; (adult men greater than or equal to 14 drinks each week and adult women greater than or equal to 7 drinks each week) 0 indicated no whereas 1 indicates yes
13. AnyHealthCare - It is represented using a 0 or 1; Coverage in the form of health insurance, a prepaid health care plan, or a health maintenance organization (HMO), etc. 0 indicates no whereas 1 indicates yes.
14. NoDocbcCost - It is represented using a 0 or 1; In the previous year, did you ever put off going to the doctor because you couldn't afford it? 0 indicates no whereas 1 indicates yes
15. GenHlth - It is represented using a 0 or 1; How would you rate the state of your health right now? on a range 1 to 5 1 is excellent 2 is very good 3 is good 4 is fair 5 is poor.
16. MentHlth - the number of days on which one's mental health has been poor, from 1 to 30.
17. PhysHlth - number of days in the past 30 spent recovering from a physical injury or sickness, ranked from 1 to 30.
18. DiffWalk - It is represented using a 0 or 1; Can you walk a long distance or climb a flight of stairs with great difficulty? 0 indicates no whereas 1 indicates yes
19. Sex - It is represented using a 0 or 1; 0 indicates female 1 indicates male
20. Age – It is divided into 13-level age category 1 equals 18 to 24 9 equals 60 to 64 13 equals 80 or older.
21. Education - Education level scale 1to 6 1 indicates Never went to school, or barely went as far as kindergarten while 2 indicates elementary etc.
22. Income - Income scale 1 to 8 1 equals less than ten thousand dollars 5 equals less than thirty five thousand dollars 8 equals seventy five thousand dollars or more.

import pandas as pd

import numpy as np

from sklearn.model\_selection import StratifiedKFold, GridSearchCV

from sklearn.tree import DecisionTreeClassifier

from sklearn.metrics import accuracy\_score

# Load the dataset

datfra = pd.read\_csv("C:/Users/ssunkari/downloads/diabetes\_binary\_5050split\_health\_indicators\_BRFSS2015.csv", encoding='latin-1')

# Check the dimensions of the dataset

print("Dataset shape: ", df.shape)

# Check for missing values and drop rows or columns with missing values

# Use the dropna() method to remove any missing values

# Explain any removed row/column and the number of missing values in it

datfra = df.dropna()

# Get the features and target columns from the dataset

San = datfra.iloc[:, :-1]

Raf = datfra.iloc[:, -1]

# Use StratifiedKFold to randomly split the data into K equal folds

strakf = StratifiedKFold(n\_splits=5, shuffle=True, random\_state=42)

# Create lists to store the accuracy scores for Gini and Entropy

gin\_scr = []

entro\_scr = []

# Create a for loop that iterates over the 5 folds

for fold, (train\_index, test\_index) in enumerate(strakf.split(San, Raf)):

print("Fold:", fold+1)

# Get the training and testing data for this fold

San\_train, San\_test = San.iloc[train\_index], San.iloc[test\_index]

Raf\_train, Raf\_test = Raf.iloc[train\_index], Raf.iloc[test\_index]

# Create a decision tree classifier

dcttree = DecisionTreeClassifier(max\_depth=X.shape[1]\*10)

# Create a parameter grid for GridSearchCV

paramet\_grd = {'criterion': ['gini', 'entropy'],

'max\_depth': [10, 20, 30]}

# Use GridSearchCV to find the best parameter values

grd\_srch = GridSearchCV(dcttree, paramet\_grd, cv=4, scoring='accuracy')

grd\_srch.fit(San\_train, Raf\_train)

# Get the best parameter values

bst\_crit = grd\_srch.best\_params\_['criterion']

bst\_dpth = grd\_srch.best\_params\_['max\_depth']

print("Best parameters:", bst\_crit, bst\_dpth)

# Train the decision tree classifier with the best parameter values

dcttree = DecisionTreeClassifier(criterion=bst\_crit, max\_depth=bst\_dpth)

dcttree.fit(San\_train, Raf\_train)

# Test the decision tree classifier and compute the accuracy score

Raf\_pred = dcttree.predict(San\_test)

accrcy = accuracy\_score(Raf\_test, Raf\_pred)

print("Accuracy:", accrcy)

# Store the accuracy score for this fold and impurity measure

if best\_criterion == 'gini':

gin\_scr.append(accrcy)

else:

entro\_scr.append(accrcy)

# Compute the overall accuracy for Gini and Entropy

gini\_accrcy = np.mean(gini\_scores)

entropy\_accrcy = np.mean(entropy\_scores)

# Print the overall accuracy for Gini and Entropy

print("Gini accuracy:", gini\_accrcy)

print("Entropy accuracy:", entropy\_accrcy)

# Determine which impurity measure gave the best results

if gini\_accrcy > entropy\_accrcy:

print("Gini gave the best results")

else:

print("Entropy gave the best results")

Graphical user interface, text, application, email

Description automatically generated

Text

Description automatically generated

Graphical user interface, text

Description automatically generated

* The entropy gave the best accuracy of 0.33338839330578124.

**References**

Alex, Teboul. (2021, November 8). *Diabetes Health Indicators Dataset.*

<https://www.kaggle.com/datasets/alexteboul/diabetes-health-indicators-dataset?resource=download&select=diabetes_binary_5050split_health_indicators_BRFSS2015.csv>