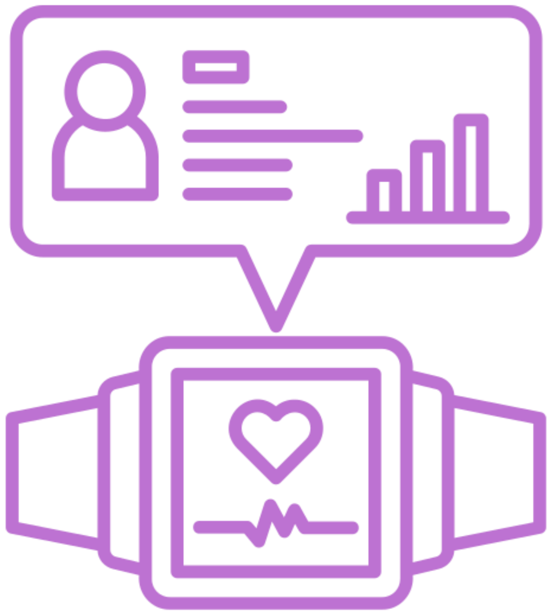
University of West Attica
Faculty of Engineering
Department of Electrical
and Electronics Engineering

Biomedical Technology



Semester Assignment

Name: Alexandros Demirtzoglou
Student ID: ee07150
Email: ee07150@uniwa.gr
Semester: 13th
Submission Date: Sun, March 17th, 2024
Submitted to: Athanasios Kiourtis

In this assignment, we will explore a practical application of the Knowledge Discovery in Databases (KDD) process and gradually apply it to a practical scenario. From data collection to implementing Python code for data assimilation, we will apply each step to our case.

Knowledge Discovery in Databases (KDD) Process:

1. Scenario Selection
2. Data Selection
3. Data Preprocessing
4. Data Transformation
5. Data Mining
6. Interpretation of Results

1. Scenario Selection

The chosen scenario is to create an algorithm for predicting heart problems by considering someone's habits. This means we aim to develop a model that can predict the likelihood of heart problems based on a person's habits and health indicators.

2. Data Selection

For this scenario, two datasets from **Kaggle.com** were selected:

- Impact of Life Factors on Heart Health
- Heart Disease Health Indicators Dataset Notebook

These datasets contain information related to lifestyle factors and heart health indicators. By analyzing these datasets, we can gain insights into the relationship between lifestyle and heart problems.

Our dataset consists of one binary target variable: HeartDiseaseorAttack, and 21 feature variables, which are either binary or ordinal. No values are missing, so the data **quality** is high, making the process easier and more **reliable**.

The column names are changed from the originals to be more readable/understandable. All variables will be explained below. Refer to the source file on which the dataset is based for the method used to clean the BRFSS2015 dataset of the Behavioral Risk Factor Surveillance System.

By analyzing the contents of the selected datasets, we find that they provide valuable information about heart health. The "Impact of Life Factors on Heart Health" dataset focuses on the impact of various lifestyle factors such as smoking, alcohol consumption, diet, physical activity, and obesity on heart disease.

On the other hand, the "Heart Disease Health Indicators" dataset includes indicators such as high blood pressure, high blood cholesterol, and smoking, identified as key risk factors for heart disease.

To apply the corresponding Data Type Definition (DTD) for this data, we must examine the characteristics and structure of the datasets. The DTD should define the data elements, their types, and any relationships between them.

Before proceeding with data mining, we need to preprocess and transform the data if necessary. This may include steps such as handling missing values, normalizing numerical variables, and encoding categorical variables. In this case, since the data quality is very high, preprocessing has been implemented by the author, as the **Usability rating** of the dataset is **10/10**. Therefore, **step 3** (Data Transformation) is being **skipped**.

Research in the field has identified the following as **significant risk factors** for heart disease and other chronic diseases such as diabetes:

- blood pressure (high)
- cholesterol (high)
- smoking
- diabetes
- obesity
- age
- sex
- race
- diet
- exercise
- alcohol consumption
- BMI
- Household Income
- Marital Status
- Sleep
- Time since last checkup
- Education
- Health care coverage
- Mental Health

(A selected subset of characteristics from the official BRFSS 2015 document.)

Let's now look in detail at what each variable means according to the author:

<u>Response Variable / Dependent Variable</u>:

- Respondents that have ever reported having coronary heart disease (CHD) or myocardial infarction (MI) --> _MICHD

<u>Independent Variables:</u>

High Blood Pressure

- Adults who have been told they have high blood pressure by a doctor, nurse, or other health professional --> _RFHYPE5

High Cholesterol

- Have you EVER been told by a doctor, nurse or other health professional that your blood cholesterol is high? --> TOLDHI2
- Cholesterol check within past five years --> _CHOLCHK

BMI

- Body Mass Index (BMI) --> _BMI5

Smoking

- Have you smoked at least 100 cigarettes in your entire life? [Note: 5 packs = 100 cigarettes] --> SMOKE100

Other Chronic Health Conditions

- (Ever told) you had a stroke. --> CVDSTRK3
- (Ever told) you have diabetes (If "Yes" and respondent is female, ask "Was this only when you were pregnant?". If Respondent says pre-diabetes or borderline diabetes, use response code 4.) --> DIABETE3

Physical Activity

- Adults who reported doing physical activity or exercise during the past 30 days other than their regular job --> _TOTINDA

Diet

- Consume Fruit 1 or more times per day --> _FRTLT1
- Consume Vegetables 1 or more times per day --> _VEGLT1

Alcohol Consumption

- Heavy drinkers (adult men having more than 14 drinks per week and adult women having more than 7 drinks per week) --> _RFDRHV5

Health Care

- Do you have any kind of health care coverage, including health insurance, prepaid plans such as HMOs, or government plans such as Medicare, or Indian Health Service? --> HLTHPLN1
- Was there a time in the past 12 months when you needed to see a doctor but could not because of cost? --> MEDCOST

Health General and Mental Health

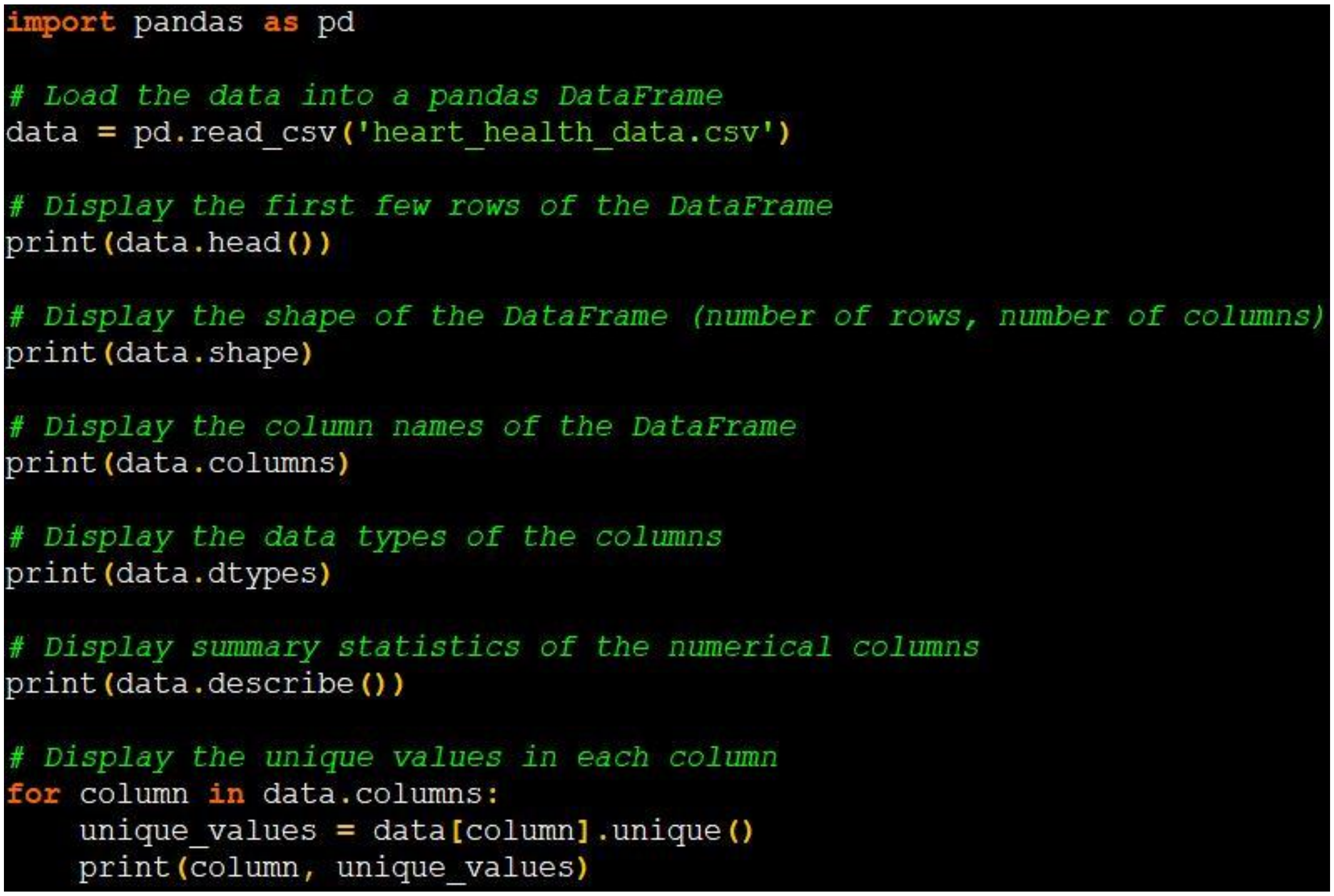
- Would you say that in general your health is: --> GENHLTH
- Now thinking about your mental health, which includes stress, depression, and problems with emotions, for how many days during the past 30 days was your mental health not good? --> MENTHLTH
- Now thinking about your physical health, which includes physical illness and injury, for how many days during the past 30 days was your physical health not good? --> PHYSHLTH
- Do you have serious difficulty walking or climbing stairs? --> DIFFWALK

Demographics

- Indicate sex of respondent. --> SEX
- Fourteen-level age category --> _AGEG5YR
- What is the highest grade or year of school you completed? --> EDUCA
- Is your annual household income from all sources: (If respondent refuses at any income level, code "Refused.") --> INCOME2

Για να αναλύσουμε τα περιεχόμενα του επιλεγμένου συνόλου δεδομένων, θα ξεκινήσουμε φορτώνοντας τα δεδομένα σε ένα **pandas DataFrame** και εξερευνώντας τη δομή και τα περιεχόμενά του.

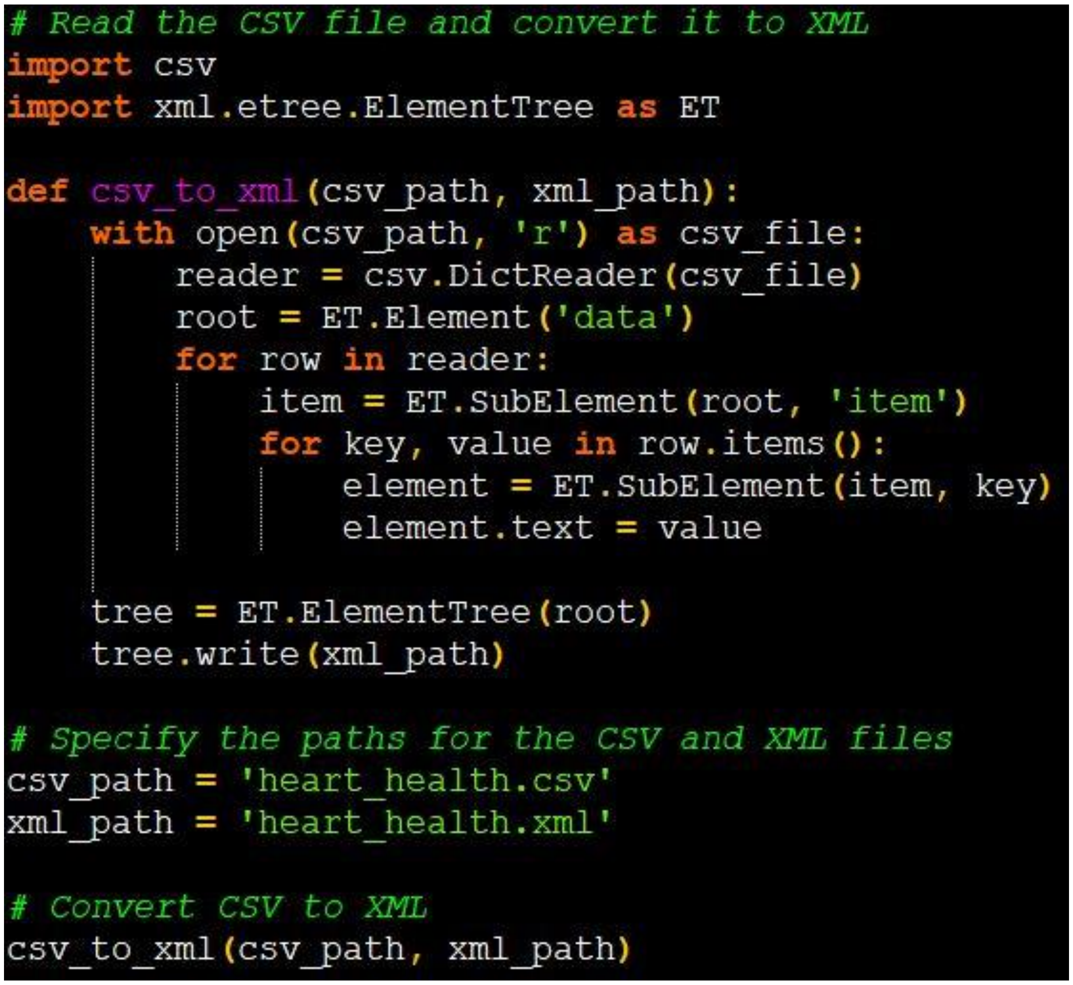
The following code block will load the data from the file **'heart_health_data.csv'** into a **pandas DataFrame** and provide various information about the data such as the first rows, the shape (number of rows and columns), column names, data types, summary statistics, and unique values in each column..



```python
import pandas as pd

# Load the data into a pandas DataFrame
data = pd.read_csv('heart_health_data.csv')

# Display the first few rows of the DataFrame
print(data.head())

# Display the shape of the DataFrame (number of rows, number of columns)
print(data.shape)

# Display the column names of the DataFrame
print(data.columns)

# Display the data types of the columns
print(data.dtypes)

# Display summary statistics of the numerical columns
print(data.describe())

# Display the unique values in each column
for column in data.columns:
    unique_values = data[column].unique()
    print(column, unique_values)
```

Next, we will see how to apply the corresponding Data Type Definition (DTD) for this data. First, we need to examine the characteristics and structure of the datasets. The DTD should define the data elements, their types, and any relationships between them..

The following code performs the **conversion** of a **CSV** file to an **XML** file. Let's examine it more closely:

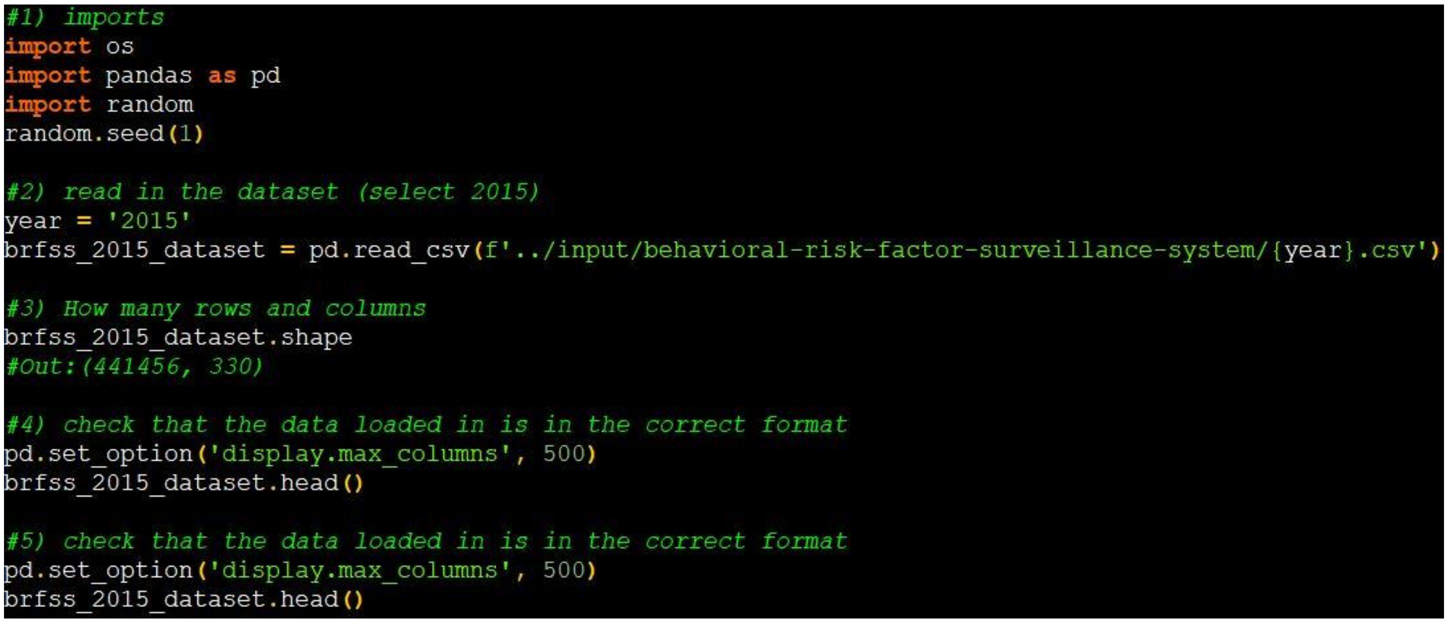


```python
# Read the CSV file and convert it to XML
import csv
import xml.etree.ElementTree as ET

def csv_to_xml(csv_path, xml_path):
    with open(csv_path, 'r') as csv_file:
        reader = csv.DictReader(csv_file)
        root = ET.Element('data')
        for row in reader:
            item = ET.SubElement(root, 'item')
            for key, value in row.items():
                element = ET.SubElement(item, key)
                element.text = value

    tree = ET.ElementTree(root)
    tree.write(xml_path)

# Specify the paths for the CSV and XML files
csv_path = 'heart_health.csv'
xml_path = 'heart_health.xml'

# Convert CSV to XML
csv_to_xml(csv_path, xml_path)
```

1. Imports the necessary **libraries**:
   - csv: Used for reading CSV files. CSV.
   - xml.etree.ElementTree as ET: This library is used for creating and editing XML data structures.
2. **Defines** a function **csv_to_xml** that takes two variables: the CSV file path and the path where the XML file will be saved.
3. **Opens** the CSV file for reading.
4. Uses csv.DictReader to **read** the CSV file and **convert** it into dictionaries, with each row being a dictionary with column names as **keys**.
5. Creates the root element of the XML tree named <**data**>.
6. Traverses each row of the CSV, **creating** a new <**item**> element for each row.
7. For each "key-value" pair in each row, creates a **new element** with the key name as the label and the value as the element's **text**.
8. Creates an **XML tree** using the root element and writes it to the specified XML file.
9. Finally, the code calls the csv_to_xml function with the **specified paths** for the CSV file and the XML file.

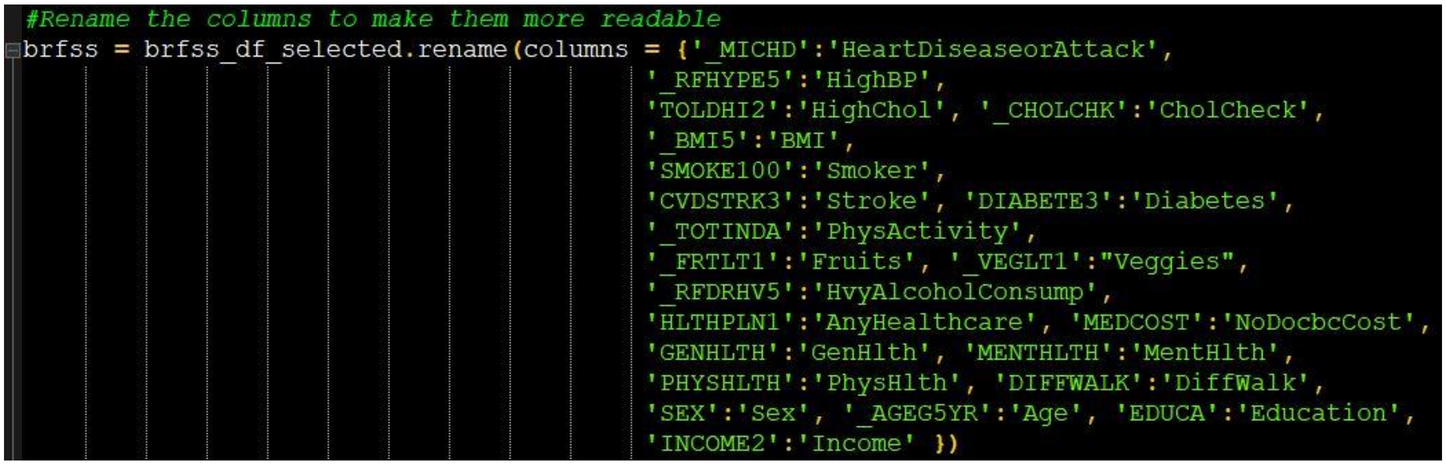
3. Data Pre-processing

In data preprocessing, we check for missing values, incorrect inputs, duplicate data, or any other imperfections in our data. As mentioned earlier, we strategically chose a dataset that is the cleaned version of the **BRFSS2015** datasheet from the organization **Behavioral Risk Factor Surveillance System** (a U.S. health research organization that examines risk factors). The quality of the data is very high, as the dataset's Usability Index has a rating of **10/10**. The following procedures were implemented to transform the source file into the cleaned file:



```
#1) imports
import os
import pandas as pd
import random
random.seed(1)

#2) read in the dataset (select 2015)
year = '2015'
brfss_2015_dataset = pd.read_csv(f'../input/behavioral-risk-factor-surveillance-system/{year}.csv')

#3) How many rows and columns
brfss_2015_dataset.shape
#Out:(441456, 330)

#4) check that the data loaded in is in the correct format
pd.set_option('display.max_columns', 500)
brfss_2015_dataset.head()

#5) check that the data loaded in is in the correct format
pd.set_option('display.max_columns', 500)
brfss_2015_dataset.head()
```

At this point, we have 441,456 rows and 330 columns. Each record contains the responses from an individual to the BRFSS survey.

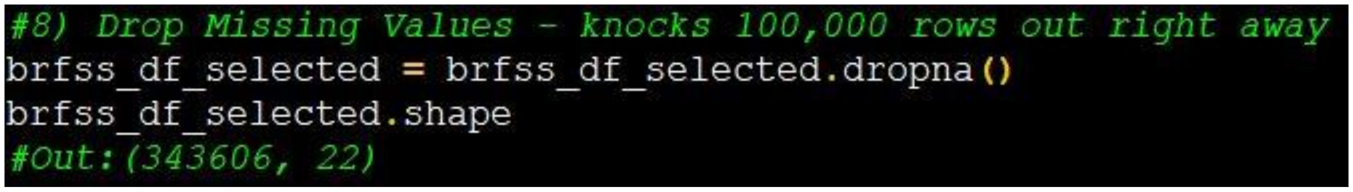


```
#Rename the columns to make them more readable
brfss = brfss_df_selected.rename(columns = {'_MICHD':'HeartDiseaseorAttack',
                                            '_RFHYPE5':'HighBP',
                                            'TOLDHI2':'HighChol', '_CHOLCHK':'CholCheck',
                                            '_BMI5':'BMI',
                                            'SMOKE100':'Smoker',
                                            'CVDSTRK3':'Stroke', 'DIABETE3':'Diabetes',
                                            '_TOTINDA':'PhysActivity',
                                            '_FRTLT1':'Fruits', '_VEGLT1':"Veggies",
                                            '_RFDRHV5':'HvyAlcoholConsump',
                                            'HLTHPLN1':'AnyHealthcare', 'MEDCOST':'NoDocbcCost',
                                            'GENHLTH':'GenHlth', 'MENTHLTH':'MentHlth',
                                            'PHYSHLTH':'PhysHlth', 'DIFFWALK':'DiffWalk',
                                            'SEX':'Sex', '_AGEG5YR':'Age', 'EDUCA':'Education',
                                            'INCOME2':'Income' })
```

4. Data Transformation

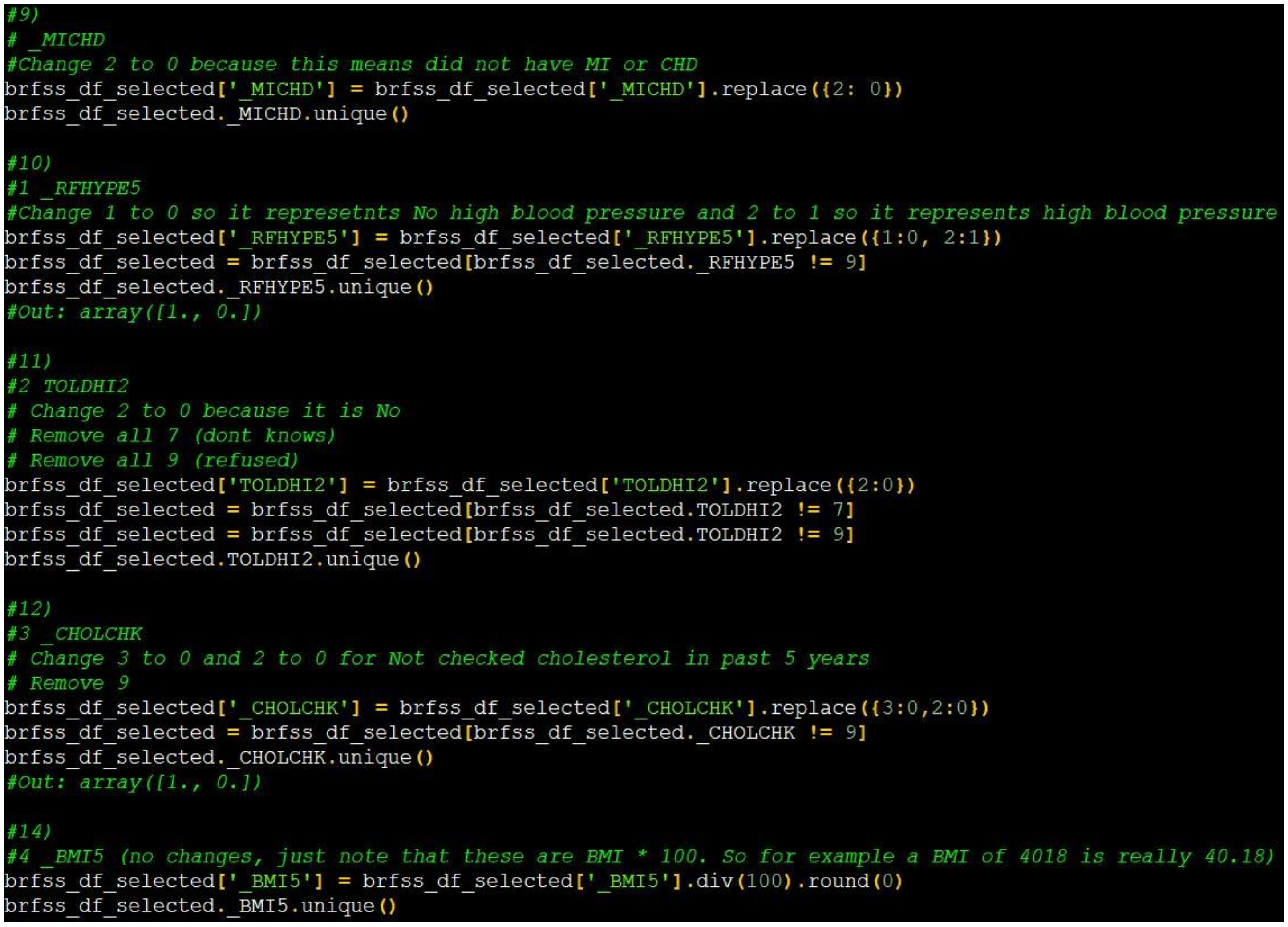
After the initial data cleaning, where only the important variables for the scenario were retained, the following summarizes some additional steps for cleaning the database. Then, we will proceed to the data mining stage, where we will analyze the data in more detail to complete the Knowledge Discovery Process.

We eliminate missing values by discarding 100,000 rows of data:



```
#8) Drop Missing Values - knocks 100,000 rows out right away
brfss_df_selected = brfss_df_selected.dropna()
brfss_df_selected.shape
#Out:(343606, 22)
```

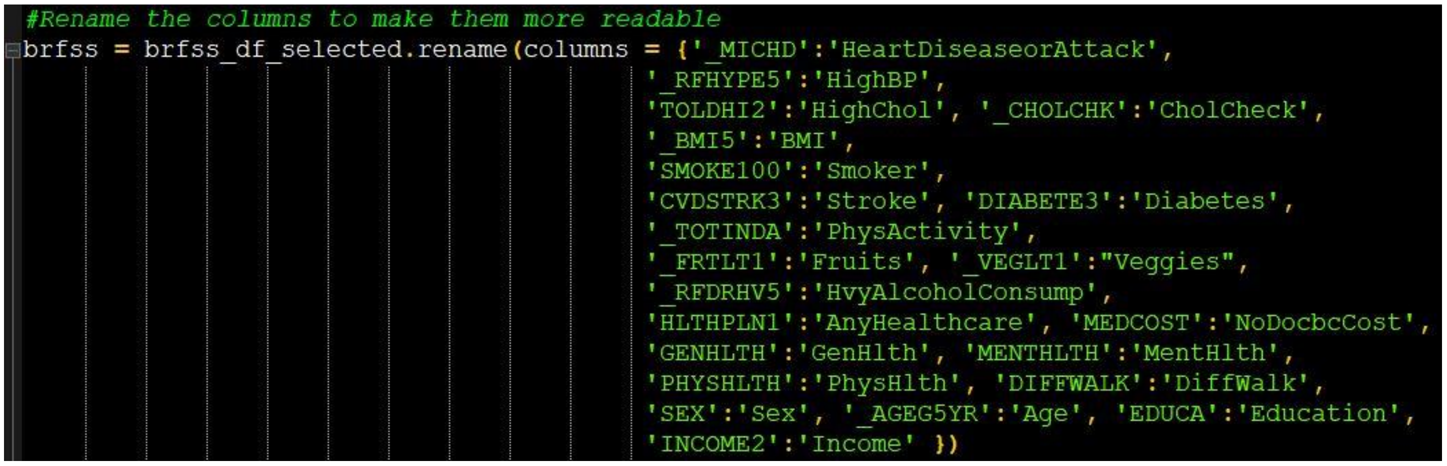
We modify and clean up values for better suitability for Machine Learning algorithms:



```
#9)
# _MICHD
#Change 2 to 0 because this means did not have MI or CHD
brfss_df_selected['_MICHD'] = brfss_df_selected['_MICHD'].replace({2: 0})
brfss_df_selected._MICHD.unique()

#10)
#1 _RFHYPE5
#Change 1 to 0 so it represetnts No high blood pressure and 2 to 1 so it represents high blood pressure
brfss_df_selected['_RFHYPE5'] = brfss_df_selected['_RFHYPE5'].replace({1:0, 2:1})
brfss_df_selected = brfss_df_selected[brfss_df_selected._RFHYPE5 != 9]
brfss_df_selected._RFHYPE5.unique()
#Out: array([1., 0.])

#11)
#2 TOLDHI2
# Change 2 to 0 because it is No
# Remove all 7 (dont knows)
# Remove all 9 (refused)
brfss_df_selected['TOLDHI2'] = brfss_df_selected['TOLDHI2'].replace({2:0})
brfss_df_selected = brfss_df_selected[brfss_df_selected.TOLDHI2 != 7]
brfss_df_selected = brfss_df_selected[brfss_df_selected.TOLDHI2 != 9]
brfss_df_selected.TOLDHI2.unique()

#12)
#3 _CHOLCHK
# Change 3 to 0 and 2 to 0 for Not checked cholesterol in past 5 years
# Remove 9
brfss_df_selected['_CHOLCHK'] = brfss_df_selected['_CHOLCHK'].replace({3:0,2:0})
brfss_df_selected = brfss_df_selected[brfss_df_selected._CHOLCHK != 9]
brfss_df_selected._CHOLCHK.unique()
#Out: array([1., 0.])

#14)
#4 _BMI5 (no changes, just note that these are BMI * 100. So for example a BMI of 4018 is really 40.18)
brfss_df_selected['_BMI5'] = brfss_df_selected['_BMI5'].div(100).round(0)
brfss_df_selected._BMI5.unique()
```

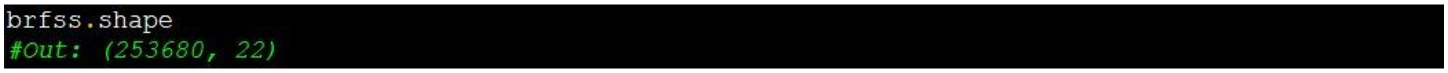
By making similar changes, we reach the final stages of data transformation, where only some aesthetic changes and the summary of the target variable remain, which is how many people have ultimately faced heart complications.

We make the feature names more readable:



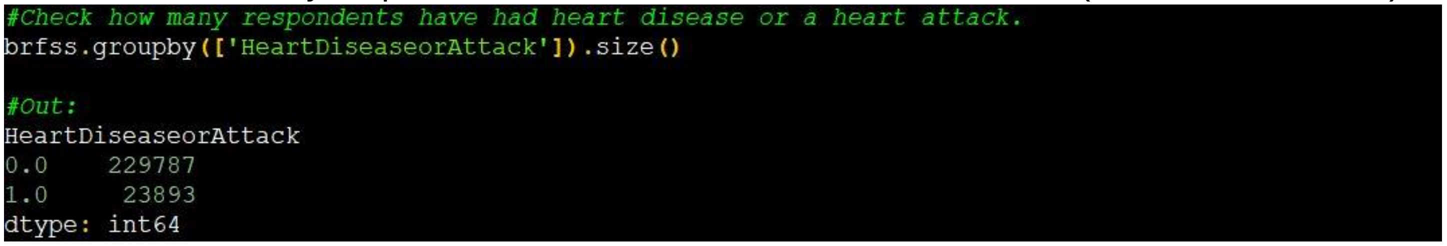
```
#Rename the columns to make them more readable
brfss = brfss_df_selected.rename(columns = {'_MICHD':'HeartDiseaseorAttack',
                                             '_RFHYPE5':'HighBP',
                                             'TOLDHI2':'HighChol', '_CHOLCHK':'CholCheck',
                                             '_BMI5':'BMI',
                                             'SMOKE100':'Smoker',
                                             'CVDSTRK3':'Stroke', 'DIABETE3':'Diabetes',
                                             '_TOTINDA':'PhysActivity',
                                             '_FRTLT1':'Fruits', '_VEGLT1':"Veggies",
                                             '_RFDRHV5':'HvyAlcoholConsump',
                                             'HLTHPLN1':'AnyHealthcare', 'MEDCOST':'NoDocbcCost',
                                             'GENHLTH':'GenHlth', 'MENTHLTH':'MentHlth',
                                             'PHYSHLTH':'PhysHlth', 'DIFFWALK':'DiffWalk',
                                             'SEX':'Sex', '_AGEG5YR':'Age', 'EDUCA':'Education',
                                             'INCOME2':'Income' })
```

From 441,456 rows and 330 columns, we now have 253,680 rows and 22 columns:



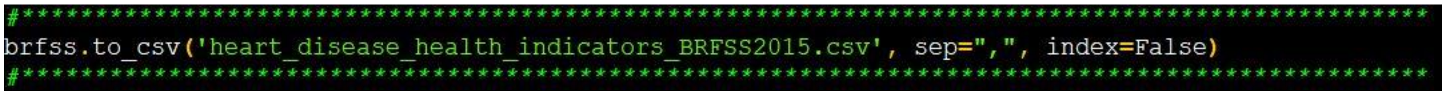
```
brfss.shape
#Out: (253680, 22)
```

We check how many respondents had heart disease or a heart attack (0.0 = No, 1.0 = Yes):



```
#Check how many respondents have had heart disease or a heart attack.
brfss.groupby(['HeartDiseaseorAttack']).size()

#Out:
HeartDiseaseorAttack
0.0    229787
1.0     23893
dtype: int64
```

Finally, we save the version where heart disease is the target variable in a .csv file and place it in the first column.



```
#*********************************************************************************************
brfss.to_csv('heart_disease_health_indicators_BRFSS2015.csv', sep=",", index=False)
#*********************************************************************************************
```

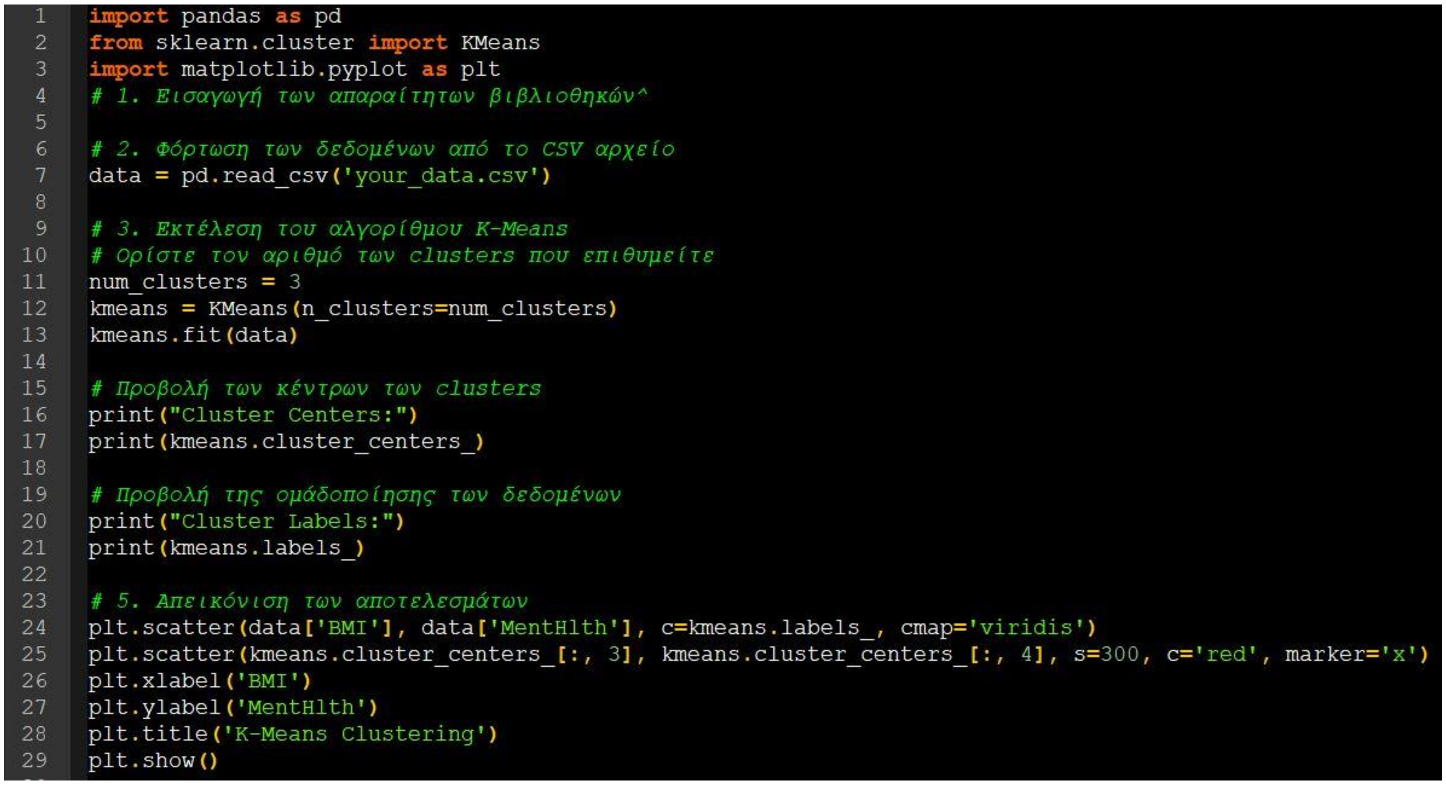
5a. Data Mining with Clustering

1. We **import** the libraries we will use, such as pandas for data management and KMeans from sklearn.cluster for the K-Means algorithm.

2. We use the pandas library to **read** the **CSV** file into a DataFrame.

3. We **set** the K-Means algorithm with the desired number of clusters (3) and train it on our data.

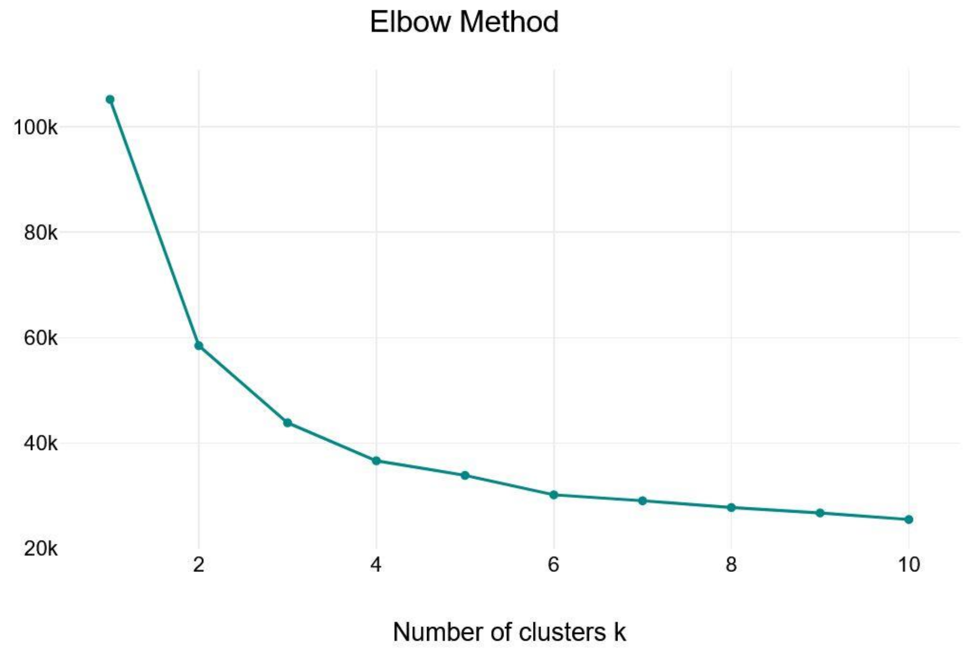
4. To **visualize** the algorithm's results, we import the matplotlib library.

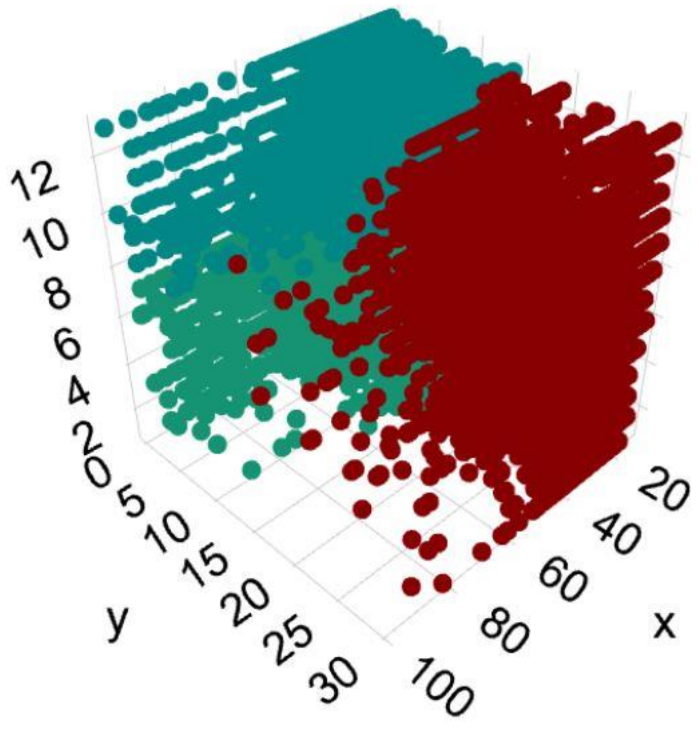


```python
import pandas as pd
from sklearn.cluster import KMeans
import matplotlib.pyplot as plt
# 1. Εισαγωγή των απαραίτητων βιβλιοθηκών^

# 2. Φόρτωση των δεδομένων από το CSV αρχείο
data = pd.read_csv('your_data.csv')

# 3. Εκτέλεση του αλγορίθμου K-Means
# Ορίστε τον αριθμό των clusters που επιθυμείτε
num_clusters = 3
kmeans = KMeans(n_clusters=num_clusters)
kmeans.fit(data)

# Προβολή των κέντρων των clusters
print("Cluster Centers:")
print(kmeans.cluster_centers_)

# Προβολή της ομάδοποίησης των δεδομένων
print("Cluster Labels:")
print(kmeans.labels_)

# 5. Απεικόνιση των αποτελεσμάτων
plt.scatter(data['BMI'], data['MentHlth'], c=kmeans.labels_, cmap='viridis')
plt.scatter(kmeans.cluster_centers_[:, 3], kmeans.cluster_centers_[:, 4], s=300, c='red', marker='x')
plt.xlabel('BMI')
plt.ylabel('MentHlth')
plt.title('K-Means Clustering')
plt.show()
```

Cluster	BMI	MentHlth	Age
1	28.12	6.81	10.13
2	30.13	24.52	7.55
3	26.34	1.42	4.87

Cluster	Cases
1	145727
2	23139
3	253434

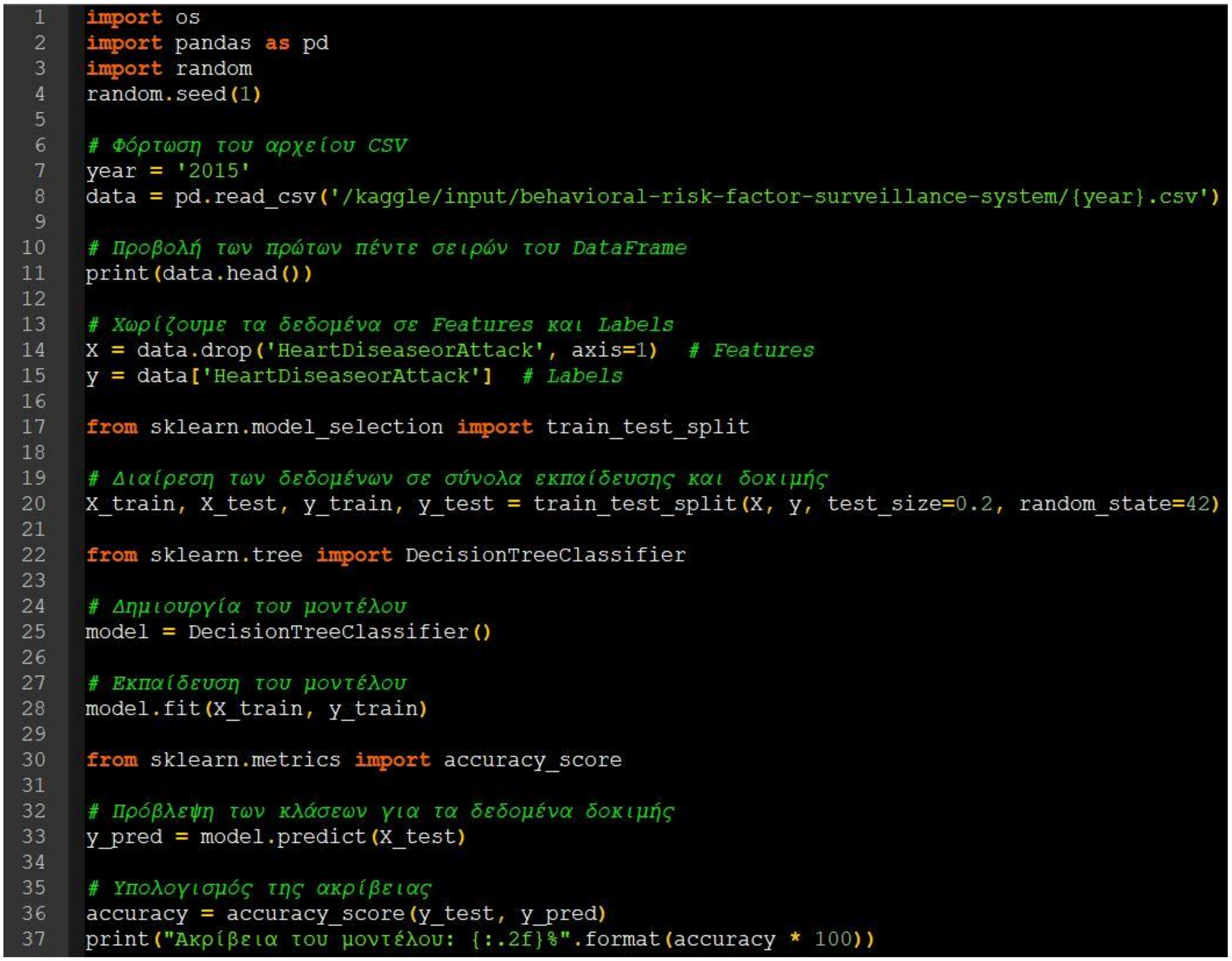




5b. Data Mining with Classification

By following the steps below, we will train a classification model using Python and the scikit-learn library.

1. First, we **load** the **.csv** file into the Python DataFrame
2. We **separate** the attributes from the column containing the classification (**target**).
3. We **split** the data into a training set and a test set.
4. We select a machine learning algorithm and **train** our model.
5. We **evaluate** our model's performance using the test set.



```python
import os
import pandas as pd
import random
random.seed(1)

# Φόρτωση του αρχείου CSV
year = '2015'
data = pd.read_csv('/kaggle/input/behavioral-risk-factor-surveillance-system/{year}.csv')

# Προβολή των πρώτων πέντε σειρών του DataFrame
print(data.head())

# Χωρίζουμε τα δεδομένα σε Features και Labels
X = data.drop('HeartDiseaseorAttack', axis=1)  # Features
y = data['HeartDiseaseorAttack']  # Labels

from sklearn.model_selection import train_test_split

# Διαίρεση των δεδομένων σε σύνολα εκπαίδευσης και δοκιμής
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

from sklearn.tree import DecisionTreeClassifier

# Δημιουργία του μοντέλου
model = DecisionTreeClassifier()

# Εκπαίδευση του μοντέλου
model.fit(X_train, y_train)

from sklearn.metrics import accuracy_score

# Πρόβλεψη των κλάσεων για τα δεδομένα δοκιμής
y_pred = model.predict(X_test)

# Υπολογισμός της ακρίβειας
accuracy = accuracy_score(y_test, y_pred)
print("Ακρίβεια του μοντέλου: {:.2f}%".format(accuracy * 100))
```

6. Interpretation of Results

Typically, when our goal is to predict a target value (e.g., the "HeartDiseaseorAttack" category), it might be more appropriate to use a classification method in data mining. However, this variable alone is quite generalized, and clustering might be more useful. Clustering can have certain advantages in our case:

Cardiovascular disease is a complex problem, and many factors associated with it may not be known in advance. The K-Means algorithm is a pattern detection algorithm in unsupervised data, meaning it can automatically discover structures and groups in our data.

Detecting hidden structures in the data can be useful in highlighting factors we may not have considered initially.

Therefore, clustering may be more suitable if we want to discover potential groups of people with similar health characteristics that may be associated with cardiovascular disease, while classification is appropriate if we have clear target labels and want to predict the likelihood of cardiovascular disease for each individual or group.

7. Bibliography

Data selection:
https://www.kaggle.com/code/alexteboul/heart-disease-health-indicators-dataset notebook?scriptVersionId=89721813
https://www.cdc.gov/brfss/annual_data/2015/pdf/codebook15_llcp.pdf

Guides & Tools:
https://www.kaggle.com/code/prashant111/k-means-clustering-with-python
https://www.convertcsv.com/csv-to-xml.htm
https://datatab.net/statistics-calculator/cluster
https://www.online-python.com/
https://drive.google.com/drive/home
https://www.geeksforgeeks.org/kdd-process-in-data-mining/
https://www.javatpoint.com/kdd-process-in-data-mining
https://agentgpt.reworkd.ai/

Icons:
https://www.flaticon.com/
https://icons8.com/

Thank you for your attention!