

# Selvakumar p

Assignment submitted on : 15.05.2020

## Code 1 : Fetching Data from csv file

Csv file used here is created for testing purpose and not from any original medical source

In [1]:

```
import pandas as pd
import numpy as np
```

In [2]:

```
train = pd.read_csv("patient.csv")
```

In [3]:

```
train.describe()
```

Out[3]:

	Patient ID	Age
count	891.000000	891.000000
mean	446.000000	31.861021
std	257.353842	15.054074
min	1.000000	0.420000
25%	223.500000	22.000000
50%	446.000000	30.000000
75%	668.500000	42.000000
max	891.000000	80.000000

In [4]:

```
train
```

Out[4]:

	Patient ID	Name	Sex	Age	Disease	Symptom1	Symptom2	Prescription	Prescription2
0	1.0	Braund, Mr. Owen Harris	male	22.0	optical	Vision problems	myopia or hypermetropia	contact lens	vitamin supplement
1	2.0	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	Cardiac	Chest Discomfort	Heart burn	Ace inhibitors	Aldosterone inhibitors
2	3.0	Heikkinen, Miss. Laina	female	26.0	obesity	fat	lack of physical activity	Physical activity	bupropion-naltrexone
3	4.0	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	obesity	fat	lack of physical activity	Physical activity	bupropion-naltrexone
4	5.0	Allen, Mr. William Henry	male	35.0	obesity	fat	lack of physical	Physical activity	bupropion-naltrexone

	Patient						activity		
	ID	Name	Sex	Age	Disease	Symptom1	Symptom2	Prescription	Prescription2
5	6.0	Moran, Mr. James	male	45.0	Cardiac	Chest Discomfort	Heart burn	Ace inhibitors	Aldosterone inhibitors
6	7.0	McCarthy, Mr. Timothy J	male	54.0	Ortho	pain in bone	Decrease in bone density	Antidepressants	Corticosteroids
7	8.0	Palsson, Master. Gosta Leonard	male	2.0	cold	Running nose	mild fever	ibuprofen	acetaminophen
8	9.0	Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)	female	27.0	obesity	fat	lack of physical activity	Physical activity	bupropion-naltrexone
9	10.0	Nasser, Mrs. Nicholas (Adele Achem)	female	14.0	fever	Temperature increase	nil	Paracetamol	ibuprofen
10	11.0	Sandstrom, Miss. Marguerite Rut	female	4.0	cold	Running nose	mild fever	ibuprofen	acetaminophen
11	12.0	Bonnell, Miss. Elizabeth	female	58.0	Ortho	pain in bone	Decrease in bone density	Antidepressants	Corticosteroids
12	13.0	Saunderscock, Mr. William Henry	male	20.0	optical	Vision problems	myopia or hypermetropia	contact lens	vitamin supplement
13	14.0	Andersson, Mr. Anders Johan	male	39.0	Cardiac	Chest Discomfort	Heart burn	Ace inhibitors	Aldosterone inhibitors
14	15.0	Vestrom, Miss. Hulda Amanda Adolfina	female	14.0	fever	Temperature increase	nil	Paracetamol	ibuprofen
15	16.0	Hewlett, Mrs. (Mary D Kingcome)	female	55.0	Ortho	pain in bone	Decrease in bone density	Antidepressants	Corticosteroids
16	17.0	Rice, Master. Eugene	male	2.0	cold	Running nose	mild fever	ibuprofen	acetaminophen
17	18.0	Williams, Mr. Charles Eugene	male	45.0	Cardiac	Chest Discomfort	Heart burn	Ace inhibitors	Aldosterone inhibitors
18	19.0	Vander Planke, Mrs. Julius (Emelia Maria Vande...	female	31.0	obesity	fat	lack of physical activity	Physical activity	bupropion-naltrexone
19	20.0	Masselmani, Mrs. Fatima	female	3.0	cold	Running nose	mild fever	ibuprofen	acetaminophen
20	21.0	Fynney, Mr. Joseph J	male	35.0	obesity	fat	lack of physical activity	Physical activity	bupropion-naltrexone
21	22.0	Beesley, Mr. Lawrence	male	34.0	obesity	fat	lack of physical activity	Physical activity	bupropion-naltrexone
22	23.0	McGowan, Miss. Anna "Annie"	female	15.0	fever	Temperature increase	nil	Paracetamol	ibuprofen
23	24.0	Sloper, Mr. William Thompson	male	28.0	obesity	fat	lack of physical activity	Physical activity	bupropion-naltrexone
24	25.0	Palsson, Miss. Torborg Danira	female	8.0	cold	Running nose	mild fever	ibuprofen	acetaminophen
25	26.0	Asplund, Mrs. Carl Oscar (Selma Augusta Emilia...	female	38.0	Cardiac	Chest Discomfort	Heart burn	Ace inhibitors	Aldosterone inhibitors
26	27.0	Emir, Mr. Farred Chehab	male	45.0	Cardiac	Chest Discomfort	Heart burn	Ace inhibitors	Aldosterone inhibitors
27	28.0	Fortune, Mr. Charles Alexander	male	19.0	optical	Vision problems	myopia or hypermetropia	contact lens	vitamin supplement
28	29.0	O'Dwyer, Miss. Ellen "Nellie"	female	45.0	Cardiac	Chest Discomfort	Heart burn	Ace inhibitors	Aldosterone inhibitors

29	800.0	Todoroff, Mr. Lallo	male	45.0	Cardiac	Chest Discomfort	Heart burn	Ace inhibitors	Aldosterone inhibitors
	Patient ID	Name	Sex	Age	Disease	Symptom1	Symptom2	Prescription	Prescription2
...	...	...	...	...	...	...	...	...	...
862	863.0	Swift, Mrs. Frederick Joel (Margaret Welles Ba...	female	48.0	Cardiac	Chest Discomfort	Heart burn	Ace inhibitors	Aldosterone inhibitors
863	864.0	Sage, Miss. Dorothy Edith "Dolly"	female	24.0	optical	Vision problems	myopia or hypermetropia	contact lens	vitamin supplement
864	865.0	Gill, Mr. John William	male	24.0	optical	Vision problems	myopia or hypermetropia	contact lens	vitamin supplement
865	866.0	Bystrom, Mrs. (Karolina)	female	42.0	Cardiac	Chest Discomfort	Heart burn	Ace inhibitors	Aldosterone inhibitors
866	867.0	Duran y More, Miss. Asuncion	female	27.0	obesity	fat	lack of physical activity	Physical activity	bupropion-naltrexone
867	868.0	Roebbling, Mr. Washington Augustus II	male	31.0	obesity	fat	lack of physical activity	Physical activity	bupropion-naltrexone
868	869.0	van Melkebeke, Mr. Philemon	male	24.0	optical	Vision problems	myopia or hypermetropia	contact lens	vitamin supplement
869	870.0	Johnson, Master. Harold Theodor	male	4.0	cold	Running nose	mild fever	ibuprofen	acetaminophen
870	871.0	Balkic, Mr. Cerin	male	26.0	obesity	fat	lack of physical activity	Physical activity	bupropion-naltrexone
871	872.0	Beckwith, Mrs. Richard Leonard (Sallie Monypeny)	female	47.0	Cardiac	Chest Discomfort	Heart burn	Ace inhibitors	Aldosterone inhibitors
872	873.0	Carlsson, Mr. Frans Olof	male	33.0	obesity	fat	lack of physical activity	Physical activity	bupropion-naltrexone
873	874.0	Vander Cruyssen, Mr. Victor	male	47.0	Cardiac	Chest Discomfort	Heart burn	Ace inhibitors	Aldosterone inhibitors
874	875.0	Abelson, Mrs. Samuel (Hannah Wizosky)	female	28.0	obesity	fat	lack of physical activity	Physical activity	bupropion-naltrexone
875	876.0	Najib, Miss. Adele Kiamie "Jane"	female	15.0	fever	Temperature increase	nil	Paracetamol	ibuproifn
876	877.0	Gustafsson, Mr. Alfred Ossian	male	20.0	optical	Vision problems	myopia or hypermetropia	contact lens	vitamin supplement
877	878.0	Petroff, Mr. Nedelio	male	19.0	optical	Vision problems	myopia or hypermetropia	contact lens	vitamin supplement
878	879.0	Laleff, Mr. Kristo	male	24.0	optical	Vision problems	myopia or hypermetropia	contact lens	vitamin supplement
879	880.0	Potter, Mrs. Thomas Jr (Lily Alexenia Wilson)	female	56.0	Ortho	pain in bone	Decrease in bone density	Antidepressants	Corticosteroids
880	881.0	Shelley, Mrs. William (Imanita Parrish Hall)	female	25.0	optical	Vision problems	myopia or hypermetropia	contact lens	vitamin supplement
881	882.0	Markun, Mr. Johann	male	33.0	obesity	fat	lack of physical activity	Physical activity	bupropion-naltrexone
882	883.0	Dahlberg, Miss. Gerda Ulrika	female	22.0	optical	Vision problems	myopia or hypermetropia	contact lens	vitamin supplement
883	884.0	Banfield, Mr. Frederick James	male	28.0	obesity	fat	lack of physical activity	Physical activity	bupropion-naltrexone
884	885.0	Sutcliffe, Mr. Henry Jr	male	25.0	optical	Vision	myopia or	contact lens	vitamin

884	885.0	Butenah, Mr. Henry Jr	male	29.0	Optical	problems	hypermetropia	contact lens	supplement
	Patient ID	Name	Sex	Age	Disease	Symptom1	Symptom2	Prescription	Prescription2
885	886.0	Rice, Mrs. William (Margaret Norton)	female	39.0	Cardiac	Chest Discomfort	Heart burn	Acc inhibitors	Aldosterone inhibitors
886	887.0	Montvila, Rev. Juozas	male	27.0	obesity	fat	lack of physical activity	Physical activity	bupropion-naltrexone
887	888.0	Graham, Miss. Margaret Edith	female	19.0	optical	Vision problems	myopia or hypermetropia	contact lens	vitamin supplement
888	889.0	Johnston, Miss. Catherine Helen "Carrie"	female	24.0	optical	Vision problems	myopia or hypermetropia	contact lens	vitamin supplement
889	890.0	Behr, Mr. Karl Howell	male	26.0	obesity	fat	lack of physical activity	Physical activity	bupropion-naltrexone
890	891.0	Dooley, Mr. Patrick	male	32.0	obesity	fat	lack of physical activity	Physical activity	bupropion-naltrexone
891	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

892 rows × 9 columns

In [5]:

```
print(train["Age"][train["Sex"] == 'female'].value_counts(normalize=True))
```

```
24.00    0.082803
35.00    0.070064
45.00    0.060510
18.00    0.041401
22.00    0.038217
30.00    0.035032
60.00    0.031847
50.00    0.028662
29.00    0.022293
19.00    0.022293
36.00    0.022293
31.00    0.022293
28.00    0.022293
21.00    0.022293
2.00     0.019108
40.00    0.019108
33.00    0.019108
16.00    0.019108
17.00    0.019108
27.00    0.019108
39.00    0.019108
38.00    0.015924
34.00    0.015924
25.00    0.015924
26.00    0.015924
4.00     0.015924
23.00    0.015924
41.00    0.012739
14.00    0.012739
9.00     0.012739
...
42.00    0.009554
3.00     0.009554
54.00    0.009554
32.00    0.009554
44.00    0.009554
8.00     0.006369
49.00    0.006369
20.00    0.006369
47.00    0.006369
1.00     0.006369
63.00    0.006369
13.00    0.006369
0.75     0.006369
6.00     0.006369
```

```
0.00      0.000369
43.00     0.006369
52.00     0.006369
14.50     0.003185
55.00     0.003185
57.00     0.003185
30.50     0.003185
51.00     0.003185
7.00      0.003185
53.00     0.003185
11.00     0.003185
62.00     0.003185
37.00     0.003185
10.00     0.003185
46.00     0.003185
32.50     0.003185
56.00     0.003185
Name: Age, Length: 64, dtype: float64
```

In [6]:

```
print(train.columns.values)
```

```
['Patient ID' 'Name' 'Sex' 'Age' 'Disease' 'Symptom1' 'Symptom2'
 'Prescription' 'Prescription2']
```

In [7]:

```
train["Age"] = train["Age"].fillna(train["Age"].median())
```

In [8]:

```
train.head()
```

Out[8]:

	Patient ID	Name	Sex	Age	Disease	Symptom1	Symptom2	Prescription	Prescription2
0	1.0	Braund, Mr. Owen Harris	male	22.0	optical	Vision problems	myopia or hypermetropia	contact lens	vitamin supplement
1	2.0	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	Cardiac	Chest Discomfort	Heart burn	Ace inhibitors	Aldosterone inhibitors
2	3.0	Heikkinen, Miss. Laina	female	26.0	obesity	fat	lack of physical activity	Physical activity	bupropion-naltrexone
3	4.0	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	obesity	fat	lack of physical activity	Physical activity	bupropion-naltrexone
4	5.0	Allen, Mr. William Henry	male	35.0	obesity	fat	lack of physical activity	Physical activity	bupropion-naltrexone

In [9]:

```
train.head(10)
```

Out[9]:

	Patient ID	Name	Sex	Age	Disease	Symptom1	Symptom2	Prescription	Prescription2
0	1.0	Braund, Mr. Owen Harris	male	22.0	optical	Vision problems	myopia or hypermetropia	contact lens	vitamin supplement
1	2.0	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	Cardiac	Chest Discomfort	Heart burn	Ace inhibitors	Aldosterone inhibitors
2	3.0	Heikkinen, Miss. Laina	female	26.0	obesity	fat	lack of physical activity	Physical activity	bupropion-naltrexone

Patient ID	Name	Sex	Age	Disease	Symptom1	Symptom2	Prescription	Prescription2
3	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	obesity	fat	lack of physical activity	Physical activity	bupropion-naltrexone
4	Allen, Mr. William Henry	male	35.0	obesity	fat	lack of physical activity	Physical activity	bupropion-naltrexone
5	Moran, Mr. James	male	45.0	Cardiac	Chest Discomfort	Heart burn	Ace inhibitors	Aldosterone inhibitors
6	McCarthy, Mr. Timothy J	male	54.0	Ortho	pain in bone	Decrease in bone density	Antidepressants	Corticosteroids
7	Palsson, Master. Gosta Leonard	male	2.0	cold	Running nose	mild fever	ibuprofen	acetaminophen
8	Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)	female	27.0	obesity	fat	lack of physical activity	Physical activity	bupropion-naltrexone
9	Nasser, Mrs. Nicholas (Adele Achem)	female	14.0	fever	Temperature increase	nil	Paracetamol	ibuprofen

## Code 2: Guess the word by 4 attempts

In [1]:

```
import random
words = ['selva', 'vp']
guesses = ''
turns = 4
word = random.choice(words)

while turns > 0:

    failed = 0

    for char in word:

        if char in guesses:
            print(char)

        else:
            print("_")

            failed += 1

    if failed == 0:

        print("You Win")

        print("The word is: ", word)
        break

    guess = input("guess a character:")
    guesses += guess

    if guess not in word:
        turns -= 1
        print("Wrong")
        print("You have", + turns, 'more guesses')
    if turns == 0:
        print("You Loose")
```

—  
—  
—  
—  
—  
guess a character:

```

guess a character:v
-
-
-
v
-
guess a character:p
Wrong
You have 3 more guesses
-
-
-
v
-
guess a character:s
s
-
-
v
-
guess a character:e
s
e
-
v
-
guess a character:l
s
e
l
v
-
guess a character:v
s
e
l
v
-
guess a character:a
s
e
l
v
a
You Win
The word is:  selva

```

## Code 3 : Chess program

In [ ]:

```

import itertools
WHITE = "white"
BLACK = "black"


class Game:

    def __init__(self):
        self.playersturn = BLACK
        self.message = "this is where prompts will go"
        self.gameboard = {}
        self.placePieces()
        print("chess program. enter moves in algebraic notation separated by space")
        self.main()

    def placePieces(self):

        for i in range(0,8):
            self.gameboard[(i,1)] = Pawn(WHITE,uniDict[WHITE][Pawn],1)
            self.gameboard[(i,6)] = Pawn(BLACK,uniDict[BLACK][Pawn],-1)

```

```

self.gameboard[(1,0)] = uniDict[BLACK, uniDict[BLACK][rank], 1, 7]

placers = [Rook,Knight,Bishop,Queen,King,Bishop,Knight,Rook]

for i in range(0,8):
    self.gameboard[(i,0)] = placers[i] (WHITE, uniDict[WHITE][placers[i]])
    self.gameboard[((7-i),7)] = placers[i] (BLACK, uniDict[BLACK][placers[i]])
placers.reverse()

def main(self):

    while True:
        self.printBoard()
        print(self.message)
        self.message = ""
        startpos,endpos = self.parseInput()
        try:
            target = self.gameboard[startpos]
        except:
            self.message = "could not find piece; index probably out of range"
            target = None

        if target:
            print("found "+str(target))
            if target.Color != self.playersturn:
                self.message = "you aren't allowed to move that piece this turn"
                continue
            if target.isValid(startpos,endpos,target.Color,self.gameboard):
                self.message = "that is a valid move"
                self.gameboard[endpos] = self.gameboard[startpos]
                del self.gameboard[startpos]
                self.isCheck()
                if self.playersturn == BLACK:
                    self.playersturn = WHITE
                else : self.playersturn = BLACK
            else :
                self.message = "invalid move" + str(target.availableMoves(startpos[0],startpos[
],self.gameboard))
                print(target.availableMoves(startpos[0],startpos[1],self.gameboard))
            else : self.message = "there is no piece in that space"

    def isCheck(self):
        #ascertain where the kings are, check all pieces of opposing color against those kings,
then if either get hit, check if its checkmate
        king = King
        kingDict = {}
        pieceDict = {BLACK : [], WHITE : []}
        for position,piece in self.gameboard.items():
            if type(piece) == King:
                kingDict[piece.Color] = position
                print(piece)
                pieceDict[piece.Color].append((piece,position))
        #white
        if self.canSeeKing(kingDict[WHITE],pieceDict[BLACK]):
            self.message = "White player is in check"
        if self.canSeeKing(kingDict[BLACK],pieceDict[WHITE]):
            self.message = "Black player is in check"

    def canSeeKing(self,kingpos,pielist):
        #checks if any pieces in piece list (which is an array of (piece,position) tuples) can see
the king in kingpos
        for piece,position in pielist:
            if piece.isValid(position,kingpos,piece.Color,self.gameboard):
                return True

    def parseInput(self):
        try:
            a,b = input().split()
            a = ((ord(a[0])-97), int(a[1])-1)
            b = (ord(b[0])-97, int(b[1])-1)
            print(a,b)
            return (a,b)
        except:
            print("error decoding input. please try again")
            return((-1,-1), (-1,-1))

```



```

    """def validateInput(self, *kargs):
        for arg in kargs:
            if type(arg[0]) is not type(1) or type(arg[1]) is not type(1):
                return False
            return True"""

def printBoard(self):
    print(" 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |")
    for i in range(0,8):
        print("-"*32)
        print(chr(i+97),end="|")
        for j in range(0,8):
            item = self.gameboard.get((i,j)," ")
            print(str(item)+' |', end = " ")
        print()
    print("-"*32)

class Piece:

    def __init__(self,color,name):
        self.name = name
        self.position = None
        self.Color = color

    def isValid(self,startpos,endpos,Color,gameboard):
        if endpos in self.availableMoves(startpos[0],startpos[1],gameboard, Color = Color):
            return True
        return False

    def __repr__(self):
        return self.name

    def __str__(self):
        return self.name

    def availableMoves(self,x,y,gameboard):
        print("ERROR: no movement for base class")

    def AdNauseum(self,x,y,gameboard, Color, intervals):

        answers = []
        for xint,yint in intervals:
            xtemp,ytemp = x+xint,y+yint
            while self.isInBounds(xtemp,ytemp):
                #print(str((xtemp,ytemp))+"is in bounds")

                target = gameboard.get((xtemp,ytemp),None)
                if target is None: answers.append((xtemp,ytemp))
                elif target.Color != Color:
                    answers.append((xtemp,ytemp))
                    break
                else:
                    break

            xtemp,ytemp = xtemp + xint,ytemp + yint
        return answers

    def isInBounds(self,x,y):

        if x >= 0 and x < 8 and y >= 0 and y < 8:
            return True
        return False

    def noConflict(self,gameboard,initialColor,x,y):

        if self.isInBounds(x,y) and ((x,y) not in gameboard) or gameboard[(x,y)].Color !=
initialColor) : return True
        return False

chessCardinals = [(1,0),(0,1),(-1,0),(0,-1)]
chessDiagonals = [(1,1),(-1,1),(1,-1),(-1,-1)]

def knightList(x,y,int1,int2):
    return [(x+int1,y+int2),(x-int1,y+int2),(x+int1,y-int2),(x-int1,y-int2),(x+int2,y+int1),(x-int2,
y+int1),(x+int2,y-int1),(x-int2,y-int1)]
def kingList(x,y):
    return [(x+1,y),(x+1,y+1),(x+1,y-1),(x,y+1),(x,y-1),(x-1,y),(x-1,y+1),(x-1,y-1)]

```

```

class Knight(Piece):
    def availableMoves(self,x,y,gameboard, Color = None):
        if Color is None : Color = self.Color
        return [(xx,yy) for xx,yy in knightList(x,y,2,1) if self.noConflict(gameboard, Color, xx, y
y)]

class Rook(Piece):
    def availableMoves(self,x,y,gameboard ,Color = None):
        if Color is None : Color = self.Color
        return self.AdNauseum(x, y, gameboard, Color, chessCardinals)

class Bishop(Piece):
    def availableMoves(self,x,y,gameboard, Color = None):
        if Color is None : Color = self.Color
        return self.AdNauseum(x, y, gameboard, Color, chessDiagonals)

class Queen(Piece):
    def availableMoves(self,x,y,gameboard, Color = None):
        if Color is None : Color = self.Color
        return self.AdNauseum(x, y, gameboard, Color, chessCardinals+chessDiagonals)

class King(Piece):
    def availableMoves(self,x,y,gameboard, Color = None):
        if Color is None : Color = self.Color
        return [(xx,yy) for xx,yy in kingList(x,y) if self.noConflict(gameboard, Color, xx, yy)]

class Pawn(Piece):
    def __init__(self,color,name,direction):
        self.name = name
        self.Color = color
        self.direction = direction
    def availableMoves(self,x,y,gameboard, Color = None):
        if Color is None : Color = self.Color
        answers = []
        if (x+1,y+self.direction) in gameboard and self.noConflict(gameboard, Color, x+1, y+self.di
rection) : answers.append((x+1,y+self.direction))
        if (x-1,y+self.direction) in gameboard and self.noConflict(gameboard, Color, x-1, y+self.di
rection) : answers.append((x-1,y+self.direction))
        if (x,y+self.direction) not in gameboard and Color == self.Color : answers.append((x,y+self
.direction))
        return answers

uniDict = {WHITE : {Pawn : "♙", Rook : "♖", Knight : "♘", Bishop : "♗", King : "♔", Queen : "♕" }, B
LACK : {Pawn : "♟", Rook : "♞", Knight : "♞", Bishop : "♝", King : "♚", Queen : "♛" }}

```

Game ()

chess program. enter moves in algebraic notation separated by space

	1	2	3	4	5	6	7	8
a	♖	♙					♙	♞
b	♙	♙					♙	♙
c	♙	♙					♙	♙
d	♗	♙					♙	♗
e	♙	♙					♙	♗
f	♙	♙					♙	♙
g	♙	♙					♙	♙
h	♖	♙					♙	♞

this is where prompts will go