LABORATORIUM 5

Zad1. Używając standardowego słownika języka Python napisać funkcje [xi, ni]=freq(x, prob=True), która dla zadanej kolumny danych x dyskretnych zwróci: unikalne wartości xi, ich estymowane prawdopodobieństwa pi lub częstości ni.

KOD NAPISANY

```python
#ZAD1
def freq(x,prob=True):
    lista={}
    for el in x:
        for xi in lista:
            if el == xi:
                break
        else:
            lista[el]=0
    for el in x:
        lista[el]+=1
    xi=list(lista.keys())
    ni=list(lista.values())
    pi=[]
    if prob:
        total=len(x)
        for n in ni:
            pi.append(n/total)
        return xi,pi
    else:
        return xi,ni

x=[1,2,2,3,1,2]
xi,pi= freq(x,prob=True)
xi1,ni= freq(x,prob=False)
print("Unikalne wartosći xi: \n", xi)
print("Częstosci ni: \n", ni)
print("Prawdopodobienstwo pi: \n", pi)
```

REZULTAT

```
Unikalne wartosći xi:
 [1, 2, 3]
Częstosci ni:
 [2, 3, 1]
Prawdopodobienstwo pi:
 [0.3333333333333333, 0.5, 0.16666666666666666]
Press any key to continue . . .
```

Zadanie 2.

Napisać funkcję [xi, yi, ni] = freq2(x,y, prob=True), która dla zadanych kolumn danych x i y zwróci: unikalne wartości atrybutów xi, yi oraz łączny rozkład częstości lub liczności ni (w zależności od prarametru prob).

KOD NAPISANY

```python
#ZAD2
def freq2(x,y, prob=True):
    lista={}
    if issparse(y):
        y=y.toarray().flatten()
    for xi,yi in zip(x,y):
        pair =(xi,yi)
        if pair not in lista:
            lista[pair]=0

    for xi,yi in zip(x,y):
        pair =(xi,yi)
        lista[pair]+=1
    xi=list(lista.keys())
    ni=list(lista.values())
    pi=[]
    if prob:
        total=len(x)
        for n in ni:
            pi.append(n/total)
        return xi,pi
    else:
        return xi,ni

x=[1,2,1,1]
y=[1,0,1,0]

xi, pi= freq2(x,y,prob=True)
xi1 ,ni= freq2(x,y,prob=False)
print("Unikalne wartosći xi: \n", xi)
print("Prawdopodobeinstwo pi: \n", pi)
print("Częstosci ni: \n", ni)
```

REZULTAT

```
Wartosci x: [1, 2, 1, 1]
Wartosci y: [1, 0, 1, 0]
Unikalne wartosći xi:
 [(1, 1), (2, 0), (1, 0)]
Prawdopodobeinstwo pi:
 [0.5, 0.25, 0.25]
Częstosci ni:
 [2, 1, 1]
Press any key to continue . . .
```

W celu wygodniejszego zobrazowania unikalnych wartości x i y funkcja zwraca listę par xi,yi a nie osobno xi i yi.

Zadanie 3

Wykorzystując powyższe funkcje, napisać funkcje, które wyliczy: entropię h=entropy(x) oraz przyrost informacji i=infogain(x,y).

```python
def entropy(x):
    _,pi=freq(x,prob=True)
    suma=0
    for p in pi:
        suma+= -p*np.log2(p)
    return suma

def infogain(x,y):
    Hx=entropy(x)
    Hy=entropy(y)
    total=len(x)
    _,pi=freq2(x,y,prob=True)
    suma=0
    for p in pi:
        suma+=-p*np.log2(p)
    return Hx+ Hy - suma

print("Wartosci x:",x)
print("Wartosci y:",y)
print("Entropia x:",entropy(x))
print("Entropia y:"entropy(y))
print("Przyrost informacji x,y",infogain(x,y))
```

```
Wartosci x: [1, 2, 1, 1]
Wartosci y: [1, 0, 1, 0]
Entropia x: 0.8112781244591328
Entropia y: 1.0
Przyrost informacji x,y 0.31127812445913294
Press any key to continue . . .
```

Zadanie 4

Wczytać dane testowe zoo.csv oraz dokona¢ selekcji/stopniowania atrybutów z wykorzystaniem kryterium przyrostu informacji

Część danych testowych zoo.csv

```
animal,hair,feathers,eggs,milk,airborne,aquatic,predator,toothed,backbone,breathes,venomous,fins,legs,tail,domestic,catsize,type
aardvark,true,false,false,true,false,false,true,true,true,true,false,false,4,false,false,true,mammal
antelope,true,false,false,true,false,false,false,true,true,true,false,false,4,true,false,true,mammal
bass,false,false,true,false,false,true,true,true,true,false,false,true,0,true,false,false,fish
bear,true,false,false,true,false,false,true,true,true,true,false,false,4,false,false,true,mammal
boar,true,false,false,true,false,false,true,true,true,true,false,false,4,true,false,true,mammal
buffalo,true,false,false,true,false,false,false,true,true,true,false,false,4,true,false,true,mammal
calf,true,false,false,true,false,false,false,true,true,true,false,false,4,true,true,true,mammal
carp,false,false,true,false,false,true,false,true,true,false,false,true,0,true,true,false,fish
catfish,false,false,true,false,false,true,true,true,true,false,false,true,0,true,false,false,fish
cavy,true,false,false,true,false,false,false,true,true,true,false,false,4,false,true,false,mammal
cheetah,true,false,false,true,false,false,true,true,true,true,false,false,4,true,false,true,mammal
```

Dokonamy porównania przyrostu informacji jaki zyskujemy dla klasy type dzięki danym na podstawie atrybutów od hair do catsize

KOD NAPISANY

```python
#ZAD4
df = pandas.read_csv("zoo.csv")
y=df["type"]

last_column=df.columns[-1]

korelacja=0;
slownik={}
for col in df.columns[1:-1]:
    korelacja=infogain(df[col],df[last_column])
    slownik[(col,last_column)]=korelacja

sorted_slownik =sorted(slownik.items(),key=lambda x: x[1], reverse=True)

for(col1,col2),korelacja in sorted_slownik:
    print(f"Korelacja między {col1} a {col2}: {korelacja}")
```

REZULTAT

```
Przyrost informacji  legs a type: 1.3630469031539394
Przyrost informacji  milk a type: 0.9743197211096901
Przyrost informacji  toothed a type: 0.8656941534932372
Przyrost informacji  eggs a type: 0.830138448363348
Przyrost informacji  hair a type: 0.7906745736101795
Przyrost informacji  feathers a type: 0.7179499765002912
Przyrost informacji  backbone a type: 0.6761627418829197
Przyrost informacji  breathes a type: 0.6144940279390556
Przyrost informacji  tail a type: 0.5004604482515029
Przyrost informacji  airborne a type: 0.4697026095047727
Przyrost informacji  fins a type: 0.46661356715038904
Przyrost informacji  aquatic a type: 0.3894874837982223
Przyrost informacji  catsize a type: 0.3084903449142815
Przyrost informacji  venomous a type: 0.1330896295351236
Przyrost informacji  predator a type: 0.093447040540831 86
Przyrost informacji  domestic a type: 0.05066877984551832
```

Zadanie 5

Sprawdzić czy funkcje freq, freq2 działają¡ dla atrybutów rzadkich (pakiet scipy.sparse). Przerobić funkcje tak aby działały dla atrybutów rzadkich.

Zmiana funkcji freg i freq2

```python
def freq(x,prob=True):
    if issparse(x):
        x=x.toarray().flatten()
    lista={}
    for el in x:
        for xi in lista:
            if el == xi:
                break
        else:
            lista[el]=0
    for el in x:
        lista[el]+=1
    xi=list(lista.keys())
    ni=list(lista.values())
    pi=[]
    if prob:
        total=len(x)
        for n in ni:
            pi.append(n/total)
        return xi,pi
    else:
        return xi,ni
```

```python
def freq2(x,y, prob=True):
    lista={}
    if issparse(x):
        x=x.toarray().flatten()
    if issparse(y):
        y=y.toarray().flatten()
    for xi,yi in zip(x,y):
        pair =(xi,yi)
        if pair not in lista:
            lista[pair]=0

    for xi,yi in zip(x,y):
        pair =(xi,yi)
        lista[pair]+=1
    xi=list(lista.keys())
    ni=list(lista.values())
    pi=[]
    if prob:
        total=len(x)
        for n in ni:
            pi.append(n/total)
        return xi,pi
    else:
        return xi,ni
```

Do funkcji dodałem dwie linijki kodu. Sprawdzam za pomocą funkcji issparse z bilioteki scipy czy zbiór danych jest macierzą rzadką. Jeśli x jest macierzą rzadką. Przekształcam macierz rzadką na jednowymiarową. Jest wiele sposobów na zmienienie tej funkcji wybrałem ten sposób, ponieważ dzięki temu funkcje mogą zachować uniwersalność, dzięki czemu działają zarówno dla list jak i dla macierzy rzadkich.

PRZETESTOWANIE FUNKCJI:

```
data_x=np.array([1,2,1,1])
data_y=np.array([1,2,1,2])
kolumny=np.array([0,1,2,3])
wiersze =np.array([0,1,1,0])

sparse_matrix_x = csr_matrix((data_x,(kolumny,wiersze)),shape=(4,2))
sparse_matrix_y = csr_matrix((data_y,(kolumny,wiersze)),shape=(4,2))

xi,pi=freq(sparse_matrix_x,prob=True)
xi,ni=freq(sparse_matrix_x,prob=False)
print("Macierz x:")
print(sparse_matrix_x)
print("Unikalne wartosći: \n", xi)
print("Częstosci ni: \n", ni)
print("Prawdopodobienstwo pi: \n", pi)

print("Macierz y:")
print(sparse_matrix_y)
xi,pi=freq2(sparse_matrix_x,sparse_matrix_y,prob=True)
xi,ni=freq2(sparse_matrix_x,sparse_matrix_y,prob=False)
print("Unikalne wartosći: \n", xi)
print("Częstosci ni: \n", ni)
print("Prawdopodobienstwo pi: \n", pi)
```

REZULTAT

```
Macierz x:
  (np.int32(0), np.int32(0))    1
  (np.int32(1), np.int32(1))    2
  (np.int32(2), np.int32(1))    1
  (np.int32(3), np.int32(0))    1
Unikalne wartosći:
 [np.int64(1), np.int64(0), np.int64(2)]
Częstosci ni:
 [3, 4, 1]
Prawdopodobienstwo pi:
 [0.375, 0.5, 0.125]
Macierz y:
  (np.int32(0), np.int32(0))    1
  (np.int32(1), np.int32(1))    2
  (np.int32(2), np.int32(1))    1
  (np.int32(3), np.int32(0))    2
Unikalne wartosći:
 [(np.int64(1), np.int64(1)), (np.int64(0), np.int64(0)), (np.int64(2), np.int64(2)), (np.int64(1), np.int64(2))]
Częstosci ni:
 [2, 4, 1, 1]
Prawdopodobienstwo pi:
 [0.25, 0.5, 0.125, 0.125]
```

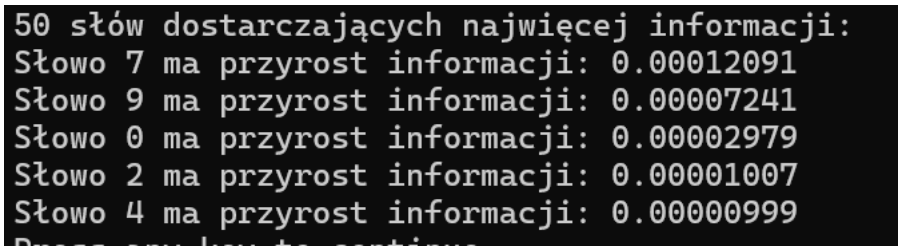
Zadanie 6.

Wykonać eksperyment podsumowujący:

KOD NAPISANY:

```
rcv1 = fetch_rcv1()
y = rcv1.target[:, 87]
x = rcv1["data"]

x.data = (x.data > 0).astype(int)

infogains = []

for i in range(50):
    word = x[:, i]
    gain = infogain(word, y)
    infogains.append((i, gain))

sorted_infogains = sorted(infogains, key=lambda x: x[1], reverse=True)

print("50 słów dostarczających najwięcej informacji:")
for i, gain in sorted_infogains[:5]:
    print(f"Słowo {i} ma przyrost informacji: {gain}")
```

REZULTAT



W ostateczności wyświetliłem tylko 5 słów dostarczających informacji, ponieważ mój komputer ma słabą moc obliczeniową i nie mógł pracować na tylu zmiennych. Nie wyświetliłem nazw słów, ponieważ ta wersja rcv1 nie ma kluczy z pełnymi nazwami słów.

PODSUMOWANIE

Wczytałem dane z korpusu Reutersa, które zawierają streszczenia artykułów prasowych oraz odpowiadające im tagi tematyczne. Następnie zbinaryzowałem dane, czyli zamieniłem liczby wystąpień słów na informację o samym ich pojawieniu się. Spośród dostępnych tagów wybrałem jeden jako zmienną decyzyjną nr 87, który oznacza artykuły sportowe. Dla każdego słowa obliczyłem przyrost informacji względem tej zmiennej decyzyjnej. Na końcu wypisałem 5 słów, które dostarczają najwięcej informacji o tym, czy artykuł należy do wybranej kategorii.