Naivný Bayes

Úvod

V mojom projekte je účelom klasifikácia jedlých a nejedlých húb na základe ich atributov.
Obsah:

1) Implementácia Bayes klasifikátora

2) Návod na použitie aplikácie založenej na tomto klasifikátore

Implementácia klasifikátora

Podľa mojej implementácie je model Bayes klasifikátora rozdelený do 3 hlavných funkcií:

- Fit
- Predict
- Test

A a tiež 2 dodatočné:

- Save
- Load

1)Fit

Hlavným účelom algoritmu klasifikátora je vytvorenie pravdepodobnostnej tabuľky na základe frekvenčnej tabuľky.

```
    for xi in range(len(X[0])):
        self.frequency_table.append(dict())
        self.likelihood_table.append(dict())
    for x in X:
        for xi in range(len(x)):
            if x[xi] in self.frequency_table[xi]:
                continue
            else:
                self.frequency_table[xi][x[xi]]= [1 for x in self.answers]
                self.likelihood_table[xi][x[xi]] = [0 for x in self.answers]
```

Obrazovka 1: Inicializácia tabuľek

Pred vyplnením tabuľky pravdepodobnosti algoritmus vyplní tabuľku frekvencií, obr.2

```
for i in range(len(X)):
    for y in range(len(X[i])):
        self.frequency_table[y][X[i][y]][Y[i]]+=1
```

Obrazovka 2: Vyplnenie tabuľky frekvencií

Potom zostavíme samotnú tabuľku pravdepodobností, obr. 3

```
for i in range(len(self.frequency_table)):
    for x in self.frequency_table[i]:
        total=sum(self.frequency_table[i][x])
        for c in range(len(self.frequency_table[i][x])):
            self.likelihood_table[i][x][c]=self.frequency_table[i][x][c]/total
```

Obrazovka 3: Vyplnenie tabuľky pravdepodobnosti

Klasifikátor bol trénovaný na datasete mushrooms.csv, https://www.kaggle.com/uciml/mushroom-classification, obr. 4

```
class,cap-shape,cap-surface,cap-color,bruises,odor,gill-attachment,gill-spacing,gill-size,gill-color,stalk-shape,stalk-root,st
p,x,s,n,t,p,f,c,n,k,e,e,s,s,w,w,p,w,o,p,k,s,u
e,x,s,y,t,a,f,c,b,k,e,c,s,s,w,w,p,w,o,p,n,n,g
e,b,s,w,t,l,f,c,b,n,e,c,s,s,w,w,p,w,o,p,n,n,m
p,x,y,w,t,p,f,c,n,n,e,e,s,s,w,w,p,w,o,p,k,s,u
e,x,s,g,f,n,f,w,b,k,t,e,s,s,w,w,p,w,o,e,n,a,g
e,x,y,y,t,a,f,c,b,n,e,c,s,s,w,w,p,w,o,p,k,n,g
e,b,s,w,t,a,f,c,b,g,e,c,s,s,w,w,p,w,o,p,k,n,m
e,b,y,w,t,l,f,c,b,n,e,c,s,s,w,w,p,w,o,p,n,s,m
p,x,y,w,t,p,f,c,n,p,e,e,s,s,w,w,p,w,o,p,k,v,g
e,b,s,y,t,a,f,c,b,g,e,c,s,s,w,w,p,w,o,p,k,s,m
e,x,y,y,t,l,f,c,b,g,e,c,s,s,w,w,p,w,o,p,n,n,g
e,x,y,y,t,a,f,c,b,n,e,c,s,s,w,w,p,w,o,p,k,s,m
e,b,s,y,t,a,f,c,b,w,e,c,s,s,w,w,p,w,o,p,n,s,g
p,x,y,w,t,p,f,c,n,k,e,e,s,s,w,w,p,w,o,p,n,v,u
e,x,f,n,f,n,f,w,b,n,t,e,s,f,w,w,p,w,o,e,k,a,g
e,s,f,g,f,n,f,c,n,k,e,e,s,s,w,w,p,w,o,p,n,y,u
e,f,f,w,f,n,f,w,b,k,t,e,s,s,w,w,p,w,o,e,n,a,g
p,x,s,n,t,p,f,c,n,n,e,e,s,s,w,w,p,w,o,p,k,s,g
p,x,y,w,t,p,f,c,n,n,e,e,s,s,w,w,p,w,o,p,n,s,u
p,x,s,n,t,p,f,c,n,k,e,e,s,s,w,w,p,w,o,p,n,s,u
e,b,s,y,t,a,f,c,b,k,e,c,s,s,w,w,p,w,o,p,n,s,m
p,x,y,n,t,p,f,c,n,n,e,e,s,s,w,w,p,w,o,p,n,v,g
```

Obrazovka 4: mushrooms.csv

2)Predict

Funkcia Predict () používa Bayesovu teorému na výpočet pravdepodobnosti príslušnosťi vstupu do každej tried, a potom pomocou funkcie argmax () vyberie triedu s najväčšou pravdepodobnosťou.

```
def predict(self, inp):
    res1=[0 for x in range(2)]
    for i in range(len(res1)):
        for y in range(len(inp)):
            if res1[i]==0:
                res1[i] = self.likelihood_table[y][inp[y]][i]
            else:
                res1[i] *= self.likelihood_table[y][inp[y]][i]

    return np.argmax(res1)
```

Obrazovka 5: predict()

3)Test

Funkcia test() nav stupe berie vstupné údaje, na základe ktorých bude model testovaný, na klasifikáciu týchto údajov používa funkciu predict() a potom vypočítava presnosť modelu porovnaním správnych odpovedí s získanými.

```
def test(self,X,Y):
    self.answers = list(set(Y))
    good=0
    all=0
    for x in range(len(X)):
        if self.predict(X[x])==Y[x]:
            good+=1
        all+=1
    accuracy=good/all
    return accuracy
```

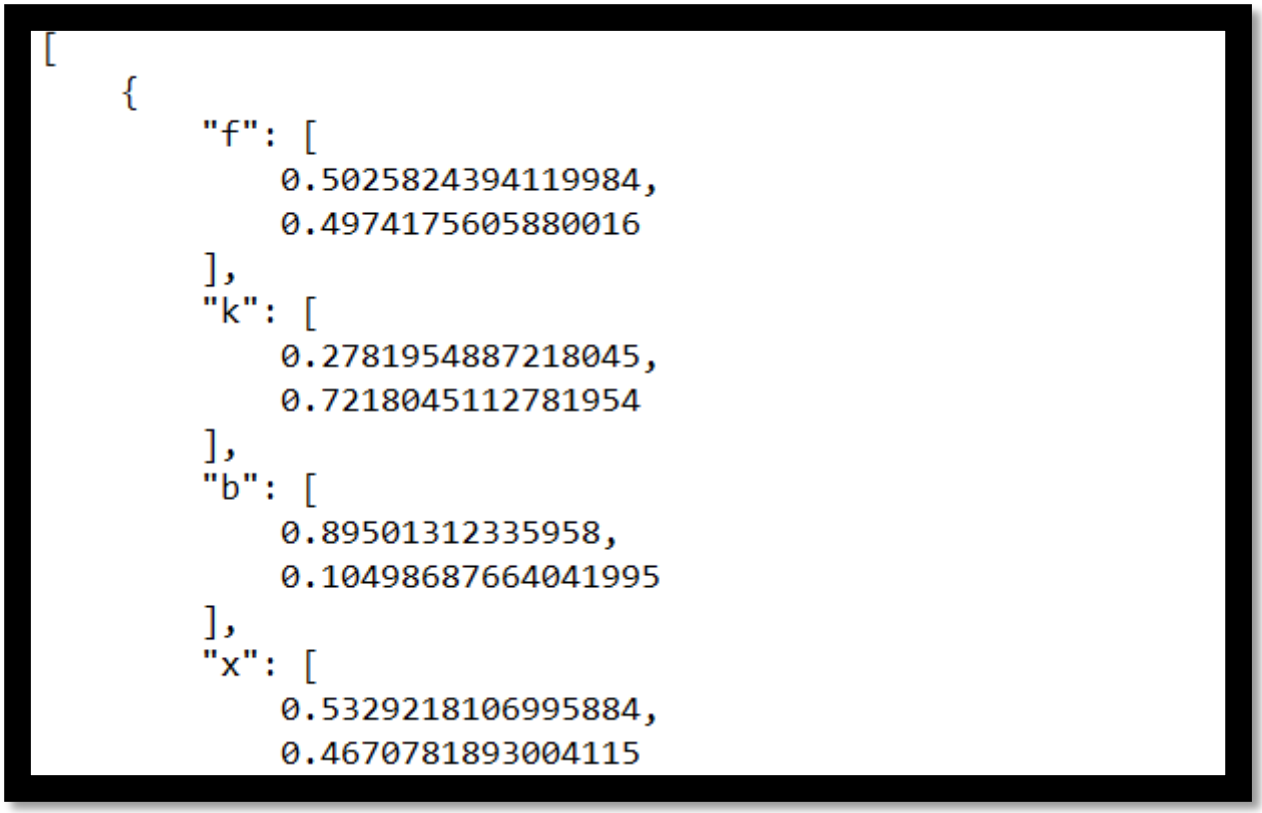
Obrazovka 6: test()

4)Save

Funkcia save () uloží tabuľku pravdepodobností získanú po učeniu ako súbor .json.

```
def save(self,name):
    with open(name+".json", 'w') as file:
        file.write(json.dumps(self.likelihood_table, indent=4))
```

Obrazovka 7: save()



Obrazovka 8: Časť uloženej tabuľky pravdepodobnosti vo formáte .json

4)Load

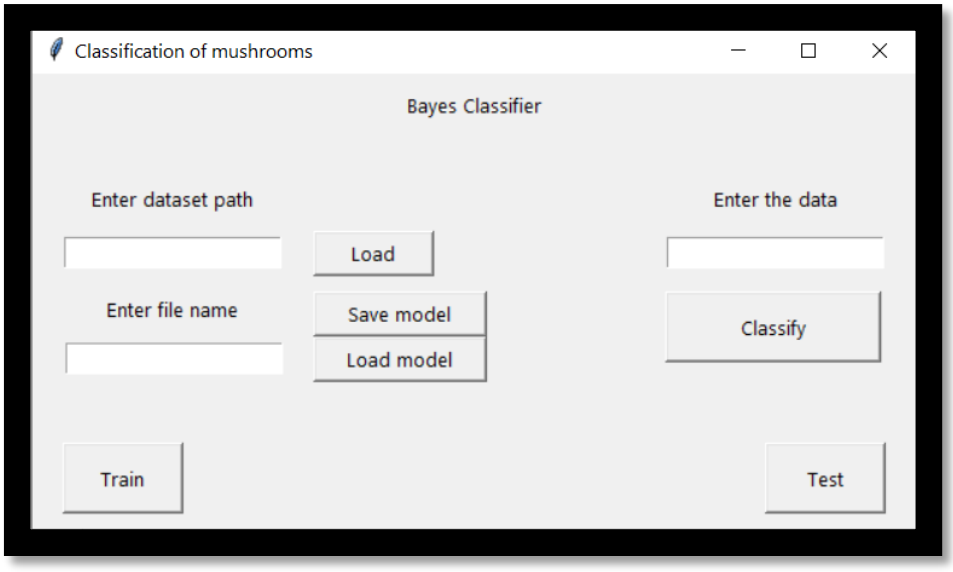
Funkcia load () načíta uloženú pravdepodobnostnú tabuľku do Bayesovho modelu, vďaka čomu nie je potreba opakovaného učenia

```
def load(self,path):
    with open(path+".json", "r") as read_file:
        self.likelihood_table = json.load(read_file)
```

Obrazovka 9: load()

Návod na použitie aplikácie

Aplikácia bola napísaná v Pythone pomocou knižnice **TKinter**, pomocou tejto aplikácie môžete trénovať Bayesov klassificator a použiť ho na klasifikáciu údajov používateľov, konkrétne na určenie typu húb (jedovatých alebo jedlých).



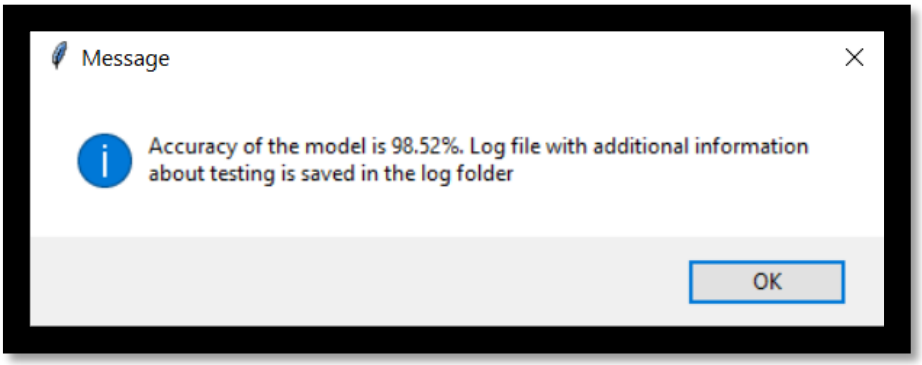
Obrazovka 10: interface aplikácií

1) Učenie a testovanie modelu

Návod na učenie modelu:

1) Do príslušného poľa zadajte cestu k datasetu a kliknite na tlačidlo **Load**
2) Stlačte tlačidlo **Train**

Pre otestovanie modelu po učeniu je potrebné kliknúť na tlačidlo **Test**, vďaka tomu sa presnosť modelu zobrazí vo vyskakovacom okne, obr. 11 a v priečinoku **Log** sa objaví log-súbor, v ktorom budú podrobnejšie informácie o testovaní Obr. 12. Ak chcete, po učeniu modelu, ho môžete uložiť zadaním názvu modelu do príslušného poľa a kliknutím na tlačidlo **Save** a potom ho načítať rovnakým spôsobom iba kliknutím na tlačidlo **Load**.



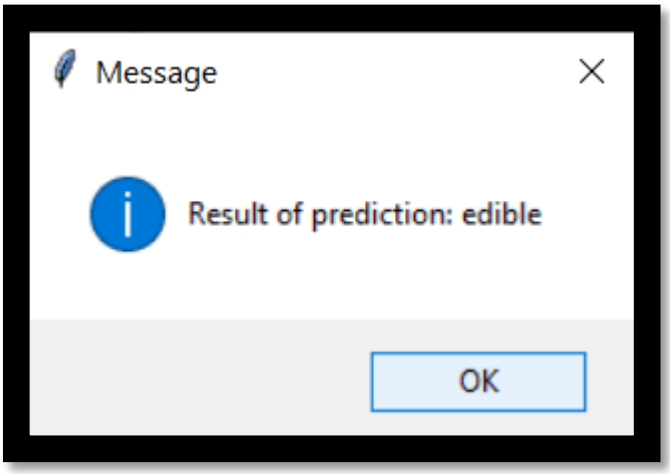
Obrazovka 11

```
2021-04-22 14:25:07.116747

Classification report:
              precision    recall  f1-score   support

           0       0.97      1.00      0.99       842
           1       1.00      0.97      0.98       783

    accuracy                           0.99      1625
   macro avg       0.99      0.98      0.99      1625
weighted avg       0.99      0.99      0.99      1625

Confusion matrix:
[840   2]
[ 22 761]
```

Obrazovka 12: Obsah log-suboru

2) Používanie modelu

Aby ste mohli klasifikovať vstupy používateľov, musíte najskôr trénovať model alebo načítať hotový, potom zadajte údaje do príslušného poľa, príklad vstupu: „b,s,w,t,l,f, c,b,n,e,c,s,s,w,w,p,w,o,p,n,n,m “. Potom kliknite na tlačidlo **Classify**, výsledok predikcie sa zobrazí v vyskakovacom okne, obr. 13.



Obrazovka 13: Výsledok predikcie