Klasifikacija_teksta

December 8, 2025

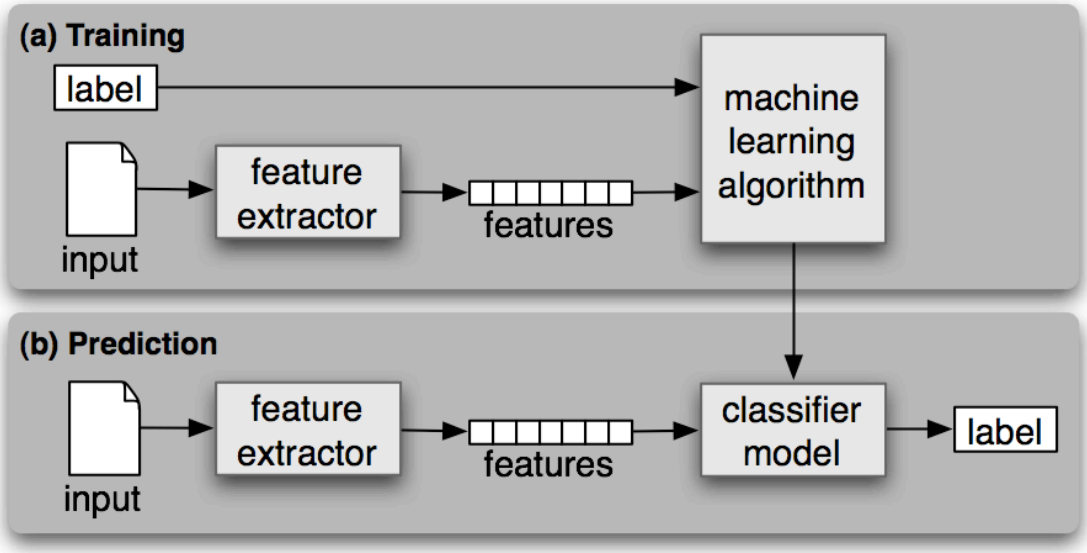
0.1 # Metode strojnog učenja u obradi teksta

[2]:
```
# preambula
# paketi
import sklearn
import nltk
import pandas as pd
import re


# scikit-learn funkcije
from sklearn.feature_extraction.text import CountVectorizer # pretvara␣
 ↪dokumente u vektore frekvencija tokena
from sklearn.feature_extraction.text import TfidfVectorizer #

# NLTK funkcije
from nltk.stem.porter import PorterStemmer # korijenovatelj engleskih riječi
from sklearn.cluster import KMeans   # algoritam klasteriranja
from pprint import pprint
```

1 Nadzirano učenje



1.1 Primjer 1: identifikacija roda

Muška i ženska imena imaju nekih bitnih razlika: imena koja završavaju na a,e,i su većim dijelom ženska imena, dok imena koja završavaju na k,o,r,s su vjerojatnije muška...

1.1.1 Klasifikator: Naivni Bayesov klasifikator

Za dani ulaz x sa svojstvima F Naivni Bayes pridružuje labelu $l \in L$ koja maksimizira vjerojatnost $P(l|F)$. Na temelju trening skupa računa

$$P(l|F) = P(F,l)/P(F)$$

gdje $P(F) = \sum_{l \in L} P(F,l)$. Za računanje $P(F,l)$ koristi prior distribuciju $P(l)$ (vjerojatnost oznake l) uz pretpostavku nezavisnosti svojstava:

$$P(F,l) = P(l) \times \prod_{f \in F} P(f|l)$$

Vjerojatnosti $P(f|l)$ i $P(l)$ računa na temelju frekvencija oznaka i svojstava u trening skupu. Npr. $P(f,l) = \text{count}(f,l)/\text{count}(l)$. U praksi, sofisticirane tehnike koriste se za zaglađivanje tih funkcija (izbjegavanje 0 vrijednosti).

Sljedeće funkcija određuje svojstvo zadnjeg slova u riječi.

[3]:
```
def gender_features(word):
    return {'last_letter':word[-1]}

gender_features('shrek')
```

[3]:
```
{'last_letter': 'k'}
```

Učitajmo sada skup podataka koji pored imena sadrži i oznake grupe.

[4]:
```
from nltk.corpus import names
import random

names = ([(name, 'male') for name in names.words('male.txt')] + [(name,␣
 ↪'female') for name in names.words('female.txt')])
random.shuffle(names)
print(names[1:10])
```

```
[('Beatrix', 'female'), ('Antonia', 'female'), ('Cori', 'female'), ('Englebert',
'male'), ('Vladamir', 'male'), ('Fanchette', 'female'), ('Carry', 'female'),
('Corbin', 'male'), ('Binky', 'male')]
```

Podijelimo `names` uz pomoć ekstraktora svojstva u 2 skupa svojstava: **skup za treniranje** i **skup za testiranje**. Iskoristimo dostupni klasifikator `NaiveBayesClassfier` (objasnit ćemo poslije…)

[5]:
```
# skup svojstava
featuresets = [(gender_features(n), g) for (n,g) in names]
# podjela u treniranje i test
train_set, test_set = featuresets[500:], featuresets[:500]
```

[6]:
```
# definiran model klasifikacije na temelju podataka
classifier = nltk.NaiveBayesClassifier.train(train_set)
```

Provjeriti klasifikator s novim pojmom. Kolika je preciznost klasifikatora?

[7]:
```
# provjeri klasifikaciju na novim pojmovima
classifier.classify(gender_features('Neo'))
classifier.classify(gender_features('Trinity'))
classifier.classify(gender_features('Cathleen'))

# provjeri preciznost na test skupu
nltk.classify.accuracy(classifier,test_set)
# koja slova najviše diskriminiraju imena?
classifier.show_most_informative_features(5)
```

```
Most Informative Features
             last_letter = 'a'            female : male   =     36.8 : 1.0
             last_letter = 'k'              male : female =     30.9 : 1.0
             last_letter = 'f'              male : female =     17.4 : 1.0
             last_letter = 'p'              male : female =     12.6 : 1.0
             last_letter = 'v'              male : female =     11.3 : 1.0
```

Problem odabira **dobrog skupa svojstava**: *overfitting* vs *underfitting*

Analiziranje loše klasifikacije možemo probati odabrati dobar skup svojstava uz pomoć **validacijskog skupa** podataka

[8]:
```
# podaci
random.shuffle(names)
train_names = names[1500:]
devtest_names = names[500:1500] # na ovom skupu radimo analizu greške
test_names = names[:500]
```

[9]:
```
# feature extractor
def gender_features(word):
    return {'last_letter':word[-1]}

# model
train_set = [(gender_features(n), g) for (n,g) in train_names]
devtest_set = [(gender_features(n), g) for (n,g) in devtest_names]
test_set = [(gender_features(n), g) for (n,g) in test_names]
# treniranje
classifier = nltk.NaiveBayesClassifier.train(train_set)
# analiza greške
print(nltk.classify.accuracy(classifier, devtest_set))
#print(nltk.classify.accuracy(classifier, test_set))

# ispitati greške
errors = []
for (name, tag) in devtest_names:
    guess = classifier.classify(gender_features(name))
    if guess != tag:
        errors.append( (tag, guess, name) )

for (tag, guess, name) in sorted(errors):
    print('correct=%-8s guess=%-8s name=%-30s' % (tag, guess, name))
```

```
0.76
correct=female   guess=male     name=Adriaens
correct=female   guess=male     name=Aidan
correct=female   guess=male     name=Aileen
correct=female   guess=male     name=Ailyn
correct=female   guess=male     name=Aleen
correct=female   guess=male     name=Alix
correct=female   guess=male     name=Alys
correct=female   guess=male     name=Ardelis
correct=female   guess=male     name=Arlen
correct=female   guess=male     name=Bab
correct=female   guess=male     name=Beret
correct=female   guess=male     name=Brear
correct=female   guess=male     name=Brynn
correct=female   guess=male     name=Carlyn
correct=female   guess=male     name=Carmen
correct=female   guess=male     name=Carol-Jean
correct=female   guess=male     name=Cass
```

```
correct=female   guess=male     name=Charleen
correct=female   guess=male     name=Chriss
correct=female   guess=male     name=Christin
correct=female   guess=male     name=Cinnamon
correct=female   guess=male     name=Clo
correct=female   guess=male     name=Clovis
correct=female   guess=male     name=Coleen
correct=female   guess=male     name=Coreen
correct=female   guess=male     name=Coriss
correct=female   guess=male     name=Denys
correct=female   guess=male     name=Devan
correct=female   guess=male     name=Devon
correct=female   guess=male     name=Dionis
correct=female   guess=male     name=Doralin
correct=female   guess=male     name=Dorcas
correct=female   guess=male     name=Eileen
correct=female   guess=male     name=Eilis
correct=female   guess=male     name=Eleanor
correct=female   guess=male     name=Elinor
correct=female   guess=male     name=Em
correct=female   guess=male     name=Emlyn
correct=female   guess=male     name=Enid
correct=female   guess=male     name=Erin
correct=female   guess=male     name=Esther
correct=female   guess=male     name=Fleur
correct=female   guess=male     name=Fran
correct=female   guess=male     name=Gilligan
correct=female   guess=male     name=Glynis
correct=female   guess=male     name=Greer
correct=female   guess=male     name=Inger
correct=female   guess=male     name=Janifer
correct=female   guess=male     name=Jess
correct=female   guess=male     name=Jo Ann
correct=female   guess=male     name=Jo-Ann
correct=female   guess=male     name=Jocelin
correct=female   guess=male     name=Jojo
correct=female   guess=male     name=Joslyn
correct=female   guess=male     name=Josselyn
correct=female   guess=male     name=Justin
correct=female   guess=male     name=Karlen
correct=female   guess=male     name=Karmen
correct=female   guess=male     name=Karon
correct=female   guess=male     name=Kerstin
correct=female   guess=male     name=Kristan
correct=female   guess=male     name=Kristyn
correct=female   guess=male     name=Kyrstin
correct=female   guess=male     name=Leanor
correct=female   guess=male     name=Linet
```

```
correct=female   guess=male     name=Lois
correct=female   guess=male     name=Lorilyn
correct=female   guess=male     name=Lust
correct=female   guess=male     name=Lynn
correct=female   guess=male     name=Madlin
correct=female   guess=male     name=Margaux
correct=female   guess=male     name=Margo
correct=female   guess=male     name=Marie-Ann
correct=female   guess=male     name=Marigold
correct=female   guess=male     name=Marijo
correct=female   guess=male     name=Meaghan
correct=female   guess=male     name=Meg
correct=female   guess=male     name=Melicent
correct=female   guess=male     name=Nan
correct=female   guess=male     name=Noreen
correct=female   guess=male     name=Pen
correct=female   guess=male     name=Robinett
correct=female   guess=male     name=Robyn
correct=female   guess=male     name=Roseann
correct=female   guess=male     name=Rozalin
correct=female   guess=male     name=Ryann
correct=female   guess=male     name=Sallyann
correct=female   guess=male     name=Sean
correct=female   guess=male     name=Shamit
correct=female   guess=male     name=Shir
correct=female   guess=male     name=Shirleen
correct=female   guess=male     name=Sioux
correct=male     guess=female   name=Ajai
correct=male     guess=female   name=Anatol
correct=male     guess=female   name=Anatole
correct=male     guess=female   name=Antony
correct=male     guess=female   name=Averell
correct=male     guess=female   name=Baillie
correct=male     guess=female   name=Barde
correct=male     guess=female   name=Barnaby
correct=male     guess=female   name=Barnie
correct=male     guess=female   name=Barthel
correct=male     guess=female   name=Bartolomei
correct=male     guess=female   name=Bjorne
correct=male     guess=female   name=Bobby
correct=male     guess=female   name=Boniface
correct=male     guess=female   name=Broddie
correct=male     guess=female   name=Burke
correct=male     guess=female   name=Chaddy
correct=male     guess=female   name=Chance
correct=male     guess=female   name=Clyde
correct=male     guess=female   name=Corby
correct=male     guess=female   name=Costa
```

```
correct=male     guess=female   name=Daffy
correct=male     guess=female   name=Danny
correct=male     guess=female   name=Darrell
correct=male     guess=female   name=Darryl
correct=male     guess=female   name=Deane
correct=male     guess=female   name=Dimitry
correct=male     guess=female   name=Doyle
correct=male     guess=female   name=Dryke
correct=male     guess=female   name=Durante
correct=male     guess=female   name=Eli
correct=male     guess=female   name=Ellsworth
correct=male     guess=female   name=Emery
correct=male     guess=female   name=Ernie
correct=male     guess=female   name=Freddy
correct=male     guess=female   name=Garth
correct=male     guess=female   name=Garvy
correct=male     guess=female   name=Gerry
correct=male     guess=female   name=Grove
correct=male     guess=female   name=Guthrey
correct=male     guess=female   name=Guthry
correct=male     guess=female   name=Hall
correct=male     guess=female   name=Hannibal
correct=male     guess=female   name=Hansel
correct=male     guess=female   name=Herbie
correct=male     guess=female   name=Hewe
correct=male     guess=female   name=Hewie
correct=male     guess=female   name=Hugh
correct=male     guess=female   name=Huntlee
correct=male     guess=female   name=Ignace
correct=male     guess=female   name=Ike
correct=male     guess=female   name=Ismail
correct=male     guess=female   name=Israel
correct=male     guess=female   name=Jake
correct=male     guess=female   name=Jean-Pierre
correct=male     guess=female   name=Jeffery
correct=male     guess=female   name=Jeffie
correct=male     guess=female   name=Jere
correct=male     guess=female   name=Jeremy
correct=male     guess=female   name=Jeromy
correct=male     guess=female   name=Jorge
correct=male     guess=female   name=Joshua
correct=male     guess=female   name=Judy
correct=male     guess=female   name=Kimball
correct=male     guess=female   name=Lanny
correct=male     guess=female   name=Laurie
correct=male     guess=female   name=Lemmy
correct=male     guess=female   name=Lesley
correct=male     guess=female   name=Lorrie
```

```
correct=male     guess=female   name=Luce
correct=male     guess=female   name=Maddie
correct=male     guess=female   name=Mahesh
correct=male     guess=female   name=Marchall
correct=male     guess=female   name=Mattie
correct=male     guess=female   name=Maury
correct=male     guess=female   name=Maxie
correct=male     guess=female   name=Merell
correct=male     guess=female   name=Merry
correct=male     guess=female   name=Michael
correct=male     guess=female   name=Mikhail
correct=male     guess=female   name=Moishe
correct=male     guess=female   name=Morrie
correct=male     guess=female   name=Moshe
correct=male     guess=female   name=Mustafa
correct=male     guess=female   name=Neil
correct=male     guess=female   name=Nevile
correct=male     guess=female   name=Niall
correct=male     guess=female   name=Oswell
correct=male     guess=female   name=Patel
correct=male     guess=female   name=Pattie
correct=male     guess=female   name=Percival
correct=male     guess=female   name=Phillipe
correct=male     guess=female   name=Pierre
correct=male     guess=female   name=Pooh
correct=male     guess=female   name=Powell
correct=male     guess=female   name=Prince
correct=male     guess=female   name=Quiggly
correct=male     guess=female   name=Ralph
correct=male     guess=female   name=Ramesh
correct=male     guess=female   name=Ramsay
correct=male     guess=female   name=Ransell
correct=male     guess=female   name=Reece
correct=male     guess=female   name=Reube
correct=male     guess=female   name=Rodge
correct=male     guess=female   name=Roni
correct=male     guess=female   name=Ronnie
correct=male     guess=female   name=Rourke
correct=male     guess=female   name=Russell
correct=male     guess=female   name=Samuele
correct=male     guess=female   name=Scarface
correct=male     guess=female   name=Sergei
correct=male     guess=female   name=Shay
correct=male     guess=female   name=Sherlocke
correct=male     guess=female   name=Sly
correct=male     guess=female   name=Solly
correct=male     guess=female   name=Steve
correct=male     guess=female   name=Tammie
```

```
correct=male     guess=female   name=Teddie
correct=male     guess=female   name=Terrell
correct=male     guess=female   name=Thane
correct=male     guess=female   name=Theodore
correct=male     guess=female   name=Thornie
correct=male     guess=female   name=Toby
correct=male     guess=female   name=Toddie
correct=male     guess=female   name=Towney
correct=male     guess=female   name=Townie
correct=male     guess=female   name=Tracey
correct=male     guess=female   name=Tray
correct=male     guess=female   name=Tulley
correct=male     guess=female   name=Tye
correct=male     guess=female   name=Udall
correct=male     guess=female   name=Udell
correct=male     guess=female   name=Uriah
correct=male     guess=female   name=Vance
correct=male     guess=female   name=Verge
correct=male     guess=female   name=Verney
correct=male     guess=female   name=Virgil
correct=male     guess=female   name=Warde
correct=male     guess=female   name=Ware
correct=male     guess=female   name=Weslie
correct=male     guess=female   name=Will
correct=male     guess=female   name=Wolfy
correct=male     guess=female   name=Woodie
correct=male     guess=female   name=Yancy
correct=male     guess=female   name=Zachariah
correct=male     guess=female   name=Zacherie
correct=male     guess=female   name=Zebadiah
correct=male     guess=female   name=Zolly
```

[10]:
```
classifier.classify(gender_features('Alys'))
```

[10]:
```
'male'
```

Što nam govori ovaj primjer? > Neki sufiksi koji imaju barem 2 člana bolje karakteriziraju ime, npr. `Cathle(en)-> female`

[11]:
```
def gender_features2(word):
    return {'suffix1':word[-1:], 'suffix2':word[-2:]}

# model
train_set = [(gender_features2(n), g) for (n,g) in train_names]
devtest_set = [(gender_features2(n), g) for (n,g) in devtest_names]
test_set = [(gender_features2(n), g) for (n,g) in test_names]
# treniranje
classifier = nltk.NaiveBayesClassifier.train(train_set)
```

```
# analiza greške
print(nltk.classify.accuracy(classifier, devtest_set))
#print(nltk.classify.accuracy(classifier, test_set))

# ispitati greške
errors = []
for (name, tag) in devtest_names:
    guess = classifier.classify(gender_features2(name))
    if guess != tag:
        errors.append( (tag, guess, name) )

for (tag, guess, name) in sorted(errors):
    print('correct=%-8s guess=%-8s name=%-30s' % (tag, guess, name))
```

```
0.777
correct=female   guess=male     name=Abagael
correct=female   guess=male     name=Adriaens
correct=female   guess=male     name=Aidan
correct=female   guess=male     name=Aileen
correct=female   guess=male     name=Aime
correct=female   guess=male     name=Aleen
correct=female   guess=male     name=Anabel
correct=female   guess=male     name=Ansley
correct=female   guess=male     name=Ardelis
correct=female   guess=male     name=Arlen
correct=female   guess=male     name=Audrey
correct=female   guess=male     name=Aurel
correct=female   guess=male     name=Avril
correct=female   guess=male     name=Bab
correct=female   guess=male     name=Beret
correct=female   guess=male     name=Brear
correct=female   guess=male     name=Brittney
correct=female   guess=male     name=Carmen
correct=female   guess=male     name=Carol-Jean
correct=female   guess=male     name=Carry
correct=female   guess=male     name=Cass
correct=female   guess=male     name=Cathy
correct=female   guess=male     name=Ceil
correct=female   guess=male     name=Charleen
correct=female   guess=male     name=Chriss
correct=female   guess=male     name=Christin
correct=female   guess=male     name=Cinnamon
correct=female   guess=male     name=Clo
correct=female   guess=male     name=Clovis
correct=female   guess=male     name=Coleen
correct=female   guess=male     name=Coreen
correct=female   guess=male     name=Corey
correct=female   guess=male     name=Coriss
```

```
correct=female   guess=male     name=Cybal
correct=female   guess=male     name=Dacey
correct=female   guess=male     name=Daniel
correct=female   guess=male     name=Delaney
correct=female   guess=male     name=Devan
correct=female   guess=male     name=Devon
correct=female   guess=male     name=Dionis
correct=female   guess=male     name=Doralin
correct=female   guess=male     name=Dorcas
correct=female   guess=male     name=Eileen
correct=female   guess=male     name=Eilis
correct=female   guess=male     name=Eleanor
correct=female   guess=male     name=Elinor
correct=female   guess=male     name=Em
correct=female   guess=male     name=Enid
correct=female   guess=male     name=Erin
correct=female   guess=male     name=Estel
correct=female   guess=male     name=Esther
correct=female   guess=male     name=Evvy
correct=female   guess=male     name=Fey
correct=female   guess=male     name=Fleur
correct=female   guess=male     name=Fran
correct=female   guess=male     name=Gabriell
correct=female   guess=male     name=Gill
correct=female   guess=male     name=Gilligan
correct=female   guess=male     name=Glynis
correct=female   guess=male     name=Greer
correct=female   guess=male     name=Haleigh
correct=female   guess=male     name=Holley
correct=female   guess=male     name=Idell
correct=female   guess=male     name=Inger
correct=female   guess=male     name=Isobel
correct=female   guess=male     name=Ivy
correct=female   guess=male     name=Janel
correct=female   guess=male     name=Janifer
correct=female   guess=male     name=Jaynell
correct=female   guess=male     name=Jess
correct=female   guess=male     name=Jocelin
correct=female   guess=male     name=Jojo
correct=female   guess=male     name=Justin
correct=female   guess=male     name=Karel
correct=female   guess=male     name=Karlen
correct=female   guess=male     name=Karmen
correct=female   guess=male     name=Karon
correct=female   guess=male     name=Kary
correct=female   guess=male     name=Kelley
correct=female   guess=male     name=Kerstin
correct=female   guess=male     name=Kial
```

```
correct=female   guess=male     name=Kourtney
correct=female   guess=male     name=Kristan
correct=female   guess=male     name=Kyrstin
correct=female   guess=male     name=Lainey
correct=female   guess=male     name=Layney
correct=female   guess=male     name=Leanor
correct=female   guess=male     name=Libbey
correct=female   guess=male     name=Linet
correct=female   guess=male     name=Linzy
correct=female   guess=male     name=Lois
correct=female   guess=male     name=Lust
correct=female   guess=male     name=Lynnell
correct=female   guess=male     name=Lynsey
correct=female   guess=male     name=Madlin
correct=female   guess=male     name=Margo
correct=female   guess=male     name=Maridel
correct=female   guess=male     name=Marigold
correct=female   guess=male     name=Marijo
correct=female   guess=male     name=Marry
correct=female   guess=male     name=May
correct=female   guess=male     name=Meaghan
correct=female   guess=male     name=Meg
correct=female   guess=male     name=Mel
correct=female   guess=male     name=Melicent
correct=female   guess=male     name=Michel
correct=female   guess=male     name=Nan
correct=female   guess=male     name=Noel
correct=female   guess=male     name=Noreen
correct=female   guess=male     name=Pen
correct=female   guess=male     name=Rachel
correct=female   guess=male     name=Robinett
correct=female   guess=male     name=Rozalin
correct=female   guess=male     name=Sean
correct=female   guess=male     name=Shamit
correct=female   guess=male     name=Shelley
correct=female   guess=male     name=Shir
correct=female   guess=male     name=Shirleen
correct=female   guess=male     name=Suzzy
correct=female   guess=male     name=Tory
correct=female   guess=male     name=Trish
correct=female   guess=male     name=Valery
correct=female   guess=male     name=Veronike
correct=female   guess=male     name=Vicky
correct=female   guess=male     name=Whitney
correct=male     guess=female   name=Ajai
correct=male     guess=female   name=Anatole
correct=male     guess=female   name=Antony
correct=male     guess=female   name=Baillie
```

```
correct=male     guess=female   name=Barde
correct=male     guess=female   name=Barnaby
correct=male     guess=female   name=Barnie
correct=male     guess=female   name=Bartolomei
correct=male     guess=female   name=Benn
correct=male     guess=female   name=Bjorne
correct=male     guess=female   name=Bobby
correct=male     guess=female   name=Boniface
correct=male     guess=female   name=Broddie
correct=male     guess=female   name=Chaddy
correct=male     guess=female   name=Chance
correct=male     guess=female   name=Clyde
correct=male     guess=female   name=Corby
correct=male     guess=female   name=Costa
correct=male     guess=female   name=Daffy
correct=male     guess=female   name=Danny
correct=male     guess=female   name=Darryl
correct=male     guess=female   name=Deane
correct=male     guess=female   name=Doyle
correct=male     guess=female   name=Durante
correct=male     guess=female   name=Eli
correct=male     guess=female   name=Ellsworth
correct=male     guess=female   name=Elwyn
correct=male     guess=female   name=Ernie
correct=male     guess=female   name=Felix
correct=male     guess=female   name=Freddy
correct=male     guess=female   name=Garth
correct=male     guess=female   name=Grove
correct=male     guess=female   name=Herbie
correct=male     guess=female   name=Hewe
correct=male     guess=female   name=Hewie
correct=male     guess=female   name=Huntlee
correct=male     guess=female   name=Ignace
correct=male     guess=female   name=Jean-Pierre
correct=male     guess=female   name=Jeffie
correct=male     guess=female   name=Jere
correct=male     guess=female   name=Jeremy
correct=male     guess=female   name=Jeromy
correct=male     guess=female   name=Jorge
correct=male     guess=female   name=Joshua
correct=male     guess=female   name=Judy
correct=male     guess=female   name=Lanny
correct=male     guess=female   name=Laurie
correct=male     guess=female   name=Lemmy
correct=male     guess=female   name=Lorrie
correct=male     guess=female   name=Luce
correct=male     guess=female   name=Maddie
correct=male     guess=female   name=Mattie
```

```
correct=male     guess=female   name=Maxie
correct=male     guess=female   name=Moishe
correct=male     guess=female   name=Morrie
correct=male     guess=female   name=Moshe
correct=male     guess=female   name=Mustafa
correct=male     guess=female   name=Nevile
correct=male     guess=female   name=Pattie
correct=male     guess=female   name=Phillipe
correct=male     guess=female   name=Pierre
correct=male     guess=female   name=Pooh
correct=male     guess=female   name=Prince
correct=male     guess=female   name=Quiggly
correct=male     guess=female   name=Reece
correct=male     guess=female   name=Reube
correct=male     guess=female   name=Rodge
correct=male     guess=female   name=Roni
correct=male     guess=female   name=Ronnie
correct=male     guess=female   name=Samuele
correct=male     guess=female   name=Scarface
correct=male     guess=female   name=Sergei
correct=male     guess=female   name=Shawn
correct=male     guess=female   name=Sly
correct=male     guess=female   name=Solly
correct=male     guess=female   name=Steve
correct=male     guess=female   name=Tammie
correct=male     guess=female   name=Teddie
correct=male     guess=female   name=Thane
correct=male     guess=female   name=Theodore
correct=male     guess=female   name=Thornie
correct=male     guess=female   name=Toby
correct=male     guess=female   name=Toddie
correct=male     guess=female   name=Townie
correct=male     guess=female   name=Tye
correct=male     guess=female   name=Uriah
correct=male     guess=female   name=Vance
correct=male     guess=female   name=Verge
correct=male     guess=female   name=Warde
correct=male     guess=female   name=Ware
correct=male     guess=female   name=Weslie
correct=male     guess=female   name=Wolfy
correct=male     guess=female   name=Woodie
correct=male     guess=female   name=Yancy
correct=male     guess=female   name=Zachariah
correct=male     guess=female   name=Zacherie
correct=male     guess=female   name=Zebadiah
correct=male     guess=female   name=Zolly
```

[12]:
```
classifier.classify(gender_features('Alys'))
```

[12]:
```
'female'
```

1.1.2 Metoda: Logistička regresija

Logistička regresija je metoda nadziranog strojnog učenja koja se koristi za binarnu klasifikaciju. Ona modelira vjerojatnost da ulazni podatak pripada određenoj klasi koristeći logističku (sigmoidnu) funkciju. Model pretvara linearnu kombinaciju ulaznih značajki u vjerojatnost između 0 i 1, što omogućava donošenje odluke o klasifikaciji na temelju praga (obično 0.5).

Matematički, logistička regresija koristi formulu:

$$P(y = 1|x) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_1 + \cdots + \beta_n x_n)}}$$

gdje su β_i koeficijenti naučeni iz podataka, a x_i ulazne značajke. Treniranje se obavlja minimizacijom funkcije gubitka, često koristeći maksimalnu vjerojatnost ili regularizirane varijante.

[ ]:
```
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction import DictVectorizer

# Priprema podataka
featuresets = [(gender_features2(n), g) for (n, g) in names]
X_dict = [f for f, g in featuresets]
y = [g for f, g in featuresets]

# pretvori rječnik u vektor (matricu)
vec = DictVectorizer()
X = vec.fit_transform(X_dict)

# Podjela na trening i test skup
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,␣
↪random_state=42)

# Treniranje logističke regresije
lr_classifier = LogisticRegression(random_state=42)
lr_classifier.fit(X_train, y_train)

# Testiranje
accuracy = lr_classifier.score(X_test, y_test)
print(f"Točnost logističke regresije: {accuracy:.2f}")

# Primjer klasifikacije
test_name = 'Alys'
test_features = vec.transform([gender_features(test_name)])
prediction = lr_classifier.predict(test_features)
print(f"Predikcija za '{test_name}': {prediction[0]}")
```

```
<Compressed Sparse Row sparse matrix of dtype 'float64'
        with 15888 stored elements and shape (7944, 313)>
  Coords        Values
  (0, 1)        1.0
  (0, 252)      1.0
  (1, 5)        1.0
  (1, 144)      1.0
  (2, 12)       1.0
  (2, 102)      1.0
  (3, 9)        1.0
  (3, 167)      1.0
  (4, 14)       1.0
  (4, 56)       1.0
  (5, 24)       1.0
  (5, 251)      1.0
  (6, 5)        1.0
  (6, 75)       1.0
  (7, 1)        1.0
  (7, 91)       1.0
  (8, 1)        1.0
  (8, 192)      1.0
  (9, 11)       1.0
  (9, 240)      1.0
  (10, 24)      1.0
  (10, 183)     1.0
  (11, 5)       1.0
  (11, 144)     1.0
  (12, 5)       1.0
  :     :
  (7931, 206)   1.0
  (7932, 9)     1.0
  (7932, 85)    1.0
  (7933, 1)     1.0
  (7933, 140)   1.0
  (7934, 1)     1.0
  (7934, 299)   1.0
  (7935, 1)     1.0
  (7935, 192)   1.0
  (7936, 13)    1.0
  (7936, 149)   1.0
  (7937, 7)     1.0
  (7937, 237)   1.0
  (7938, 24)    1.0
  (7938, 203)   1.0
  (7939, 1)     1.0
  (7939, 192)   1.0
  (7940, 1)     1.0
  (7940, 252)   1.0
```

```
  (7941, 1)     1.0
  (7941, 160)   1.0
  (7942, 9)     1.0
  (7942, 238)   1.0
  (7943, 5)     1.0
  (7943, 144)   1.0
Točnost logističke regresije: 0.79
Predikcija za 'Alys': male
```

2 Nenadzirano učenje

2.1 Analiziranje teksta

Pristup analize: *rečenica kao* ***neporedani skup riječi*** *("bag-of-words" analiza).* Riječi se vektoriziraju kao numeričke vrijednosti prema koracima: 1. **Raščlamba (tokenizacija)**: pretvara riječi/fraze u niz tokena i pridružuje numeričke identifikatore; `nltk.tokenize` 2. **Prebrojavanje**: prebrojavanje pojave tokena u dokumentima `scikit-learn` 3. **Normalizacija**: ponderiranje važnosti tokena u dokumentima `scikit-learn`

Reprezentacija teksta u računalu: * frekvencija pojedinog tokena u dokumentu: **svojstvo (engl. feature)**: * vektor frekvencija tokena: **uzorak** * koje **statistike** u postupku normalizacije reflektiraju koliko je važna neka riječ u dokumentu nekog korpusa? TF-IDF

> **CILJ:** Odrediti sličnost dokumenata

Primjer vlastitog korpusa dokumenata:

[14]:
```
# primjer teksta
from data import texts1 as texts
pprint(texts)
```

```
['Penny bought bright blue fishes.',
 'Penny bought bright blue and orange fish.',
 'The cat ate a fish at the store.',
 'Penny went to the store. Penny ate a bug. Penny saw a fish.',
 'It meowed once at the bug, it is still meowing at the bug and the fish',
 'The cat is at the fish store. The cat is orange. The cat is meowing at the '
 'fish.',
 'Penny is a fish']
```

Vektorizirajmo dokumente s obzirom na pojavnost tokena uz pomoć `sklearn`-ova `CountVector`

[15]:
```
count_vectorizer = CountVectorizer()
# tokeniziraj i prebroj
X = count_vectorizer.fit_transform(texts)
# koristi panadas za prikaz
#pd.DataFrame(X.toarray())
# poboljsaj prikaz?
pd.DataFrame(X.toarray(),columns=count_vectorizer.get_feature_names_out())
```

[15]:

	and	at	ate	blue	bought	bright	bug	cat	fish	fishes	…	meowing	\
0	0	0	0	1	1	1	0	0	0	1	…	0	
1	1	0	0	1	1	1	0	0	1	0	…	0	
2	0	1	1	0	0	0	0	1	1	0	…	0	
3	0	0	1	0	0	0	1	0	1	0	…	0	
4	1	2	0	0	0	0	2	0	1	0	…	1	
5	0	2	0	0	0	0	0	3	2	0	…	1	
6	0	0	0	0	0	0	0	0	1	0	…	0	

	once	orange	penny	saw	still	store	the	to	went
0	0	0	1	0	0	0	0	0	0
1	0	1	1	0	0	0	0	0	0
2	0	0	0	0	0	1	2	0	0
3	0	0	3	1	0	1	1	1	1
4	1	0	0	0	1	0	3	0	0
5	0	1	0	0	0	1	5	0	0
6	0	0	1	0	0	0	0	0	0

[7 rows x 23 columns]

Iz prethodnog vidimo da retci odgovaraju našim dokumentima, a stupci riječima unutar rečenice. Uočimo da u tekstu ima podosta **učestalih riječi** (engl. stopwords) koje želimo filtrirati!

[16]:
```
# novi vektorizer
count_vectorizer = CountVectorizer(stop_words='english')
X = count_vectorizer.fit_transform(texts)
pd.DataFrame(X.toarray(),columns=count_vectorizer.get_feature_names_out())
```

[16]:

	ate	blue	bought	bright	bug	cat	fish	fishes	meowed	meowing	orange	\
0	0	1	1	1	0	0	0	1	0	0	0	
1	0	1	1	1	0	0	1	0	0	0	1	
2	1	0	0	0	0	1	1	0	0	0	0	
3	1	0	0	0	1	0	1	0	0	0	0	
4	0	0	0	0	2	0	1	0	1	1	0	
5	0	0	0	0	0	3	2	0	0	1	1	
6	0	0	0	0	0	0	1	0	0	0	0	

	penny	saw	store	went
0	1	0	0	0
1	1	0	0	0
2	0	0	1	0
3	3	1	1	1
4	0	0	0	0
5	0	0	1	0
6	1	0	0	0

Dodatna filtracija tokena: > normalizacija (lematizacija ili korijenovanje) tokena.

[17]:
```
from nltk.stem import WordNetLemmatizer

# lematizator
lemmatizer = WordNetLemmatizer()
# tokenizator
def tokenizer(text):
    words = re.sub(r'[^A-Za-z0-9\-]'," ",text).lower().split() # prave riječi u␣
 ↪tekstu
    words = [lemmatizer.lemmatize(word) for word in words]
    return words
# vektorizator
count_vectorizer = CountVectorizer(stop_words='english',tokenizer=tokenizer)
X = count_vectorizer.fit_transform(texts)
pd.DataFrame(X.toarray(),columns=count_vectorizer.get_feature_names_out())
```

```
C:\Users\Domagoj\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.13_qbz
5n2kfra8p0\LocalCache\local-packages\Python313\site-
packages\sklearn\feature_extraction\text.py:517: UserWarning: The parameter
'token_pattern' will not be used since 'tokenizer' is not None'
  warnings.warn(
C:\Users\Domagoj\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.13_qbz
5n2kfra8p0\LocalCache\local-packages\Python313\site-
packages\sklearn\feature_extraction\text.py:402: UserWarning: Your stop_words
may be inconsistent with your preprocessing. Tokenizing the stop words generated
tokens ['ha', 'u', 'wa'] not in stop_words.
  warnings.warn(
C:\Users\Domagoj\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.13_qbz
5n2kfra8p0\LocalCache\local-packages\Python313\site-
packages\sklearn\feature_extraction\text.py:402: UserWarning: Your stop_words
may be inconsistent with your preprocessing. Tokenizing the stop words generated
tokens ['ha', 'u', 'wa'] not in stop_words.
  warnings.warn(
```

[17]:

	ate	blue	bought	bright	bug	cat	fish	meowed	meowing	orange	penny	saw	store	went
0	0	1	1	1	0	0	1	0	0	0	1	0	0	0
1	0	1	1	1	0	0	1	0	0	1	1	0	0	0
2	1	0	0	0	0	1	1	0	0	0	0	0	1	0
3	1	0	0	0	1	0	1	0	0	0	3	1	1	1
4	0	0	0	0	2	0	1	1	1	0	0			
5	0	0	0	0	0	3	2	0	1	1	0			
6	0	0	0	0	0	0	1	0	0	0	1			

```
4    0       0      0
5    0       1      0
6    0       0      0
```

TF-IDF Term-Frequency-Inverse-Document-Frequency je statistika koja se koristi za vrednovanje koliko neka riječ nosi informacije u korpusu s obzirom na njenu pojavnost:

$$\text{tf-idf}(t, d) = \text{tf}(t, d) \times \text{idf}(t)$$

Term Frequency (Frekvencija pojma) Utvrđivanje relativne pojavnosti tokena t u dokumentu d:

$$\text{tf}(t, d) = \frac{f(t, d)}{\sum_{v \in d} f(v, d)}$$

gfje je $f(v, d)$ broj pojave tokena v u dokumentu d.

Koristimo `sklearn`-ov `TfidfVectorizer`.

[18]:
```
tfidf_vectorizer = TfidfVectorizer(stop_words='english', tokenizer=tokenizer,␣
↪use_idf=False, norm='l1')
X = tfidf_vectorizer.fit_transform(texts)
df=pd.DataFrame(X.toarray(), columns = tfidf_vectorizer.get_feature_names_out())
pprint(texts)
df
```

```
['Penny bought bright blue fishes.',
 'Penny bought bright blue and orange fish.',
 'The cat ate a fish at the store.',
 'Penny went to the store. Penny ate a bug. Penny saw a fish.',
 'It meowed once at the bug, it is still meowing at the bug and the fish',
 'The cat is at the fish store. The cat is orange. The cat is meowing at the '
 'fish.',
 'Penny is a fish']
```

```
C:\Users\Domagoj\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.13_qbz
5n2kfra8p0\LocalCache\local-packages\Python313\site-
packages\sklearn\feature_extraction\text.py:517: UserWarning: The parameter
'token_pattern' will not be used since 'tokenizer' is not None'
  warnings.warn(
C:\Users\Domagoj\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.13_qbz
5n2kfra8p0\LocalCache\local-packages\Python313\site-
packages\sklearn\feature_extraction\text.py:402: UserWarning: Your stop_words
may be inconsistent with your preprocessing. Tokenizing the stop words generated
tokens ['ha', 'u', 'wa'] not in stop_words.
  warnings.warn(
```

[18]:

	ate	blue	bought	bright	bug	cat	fish	meowed	\
0	0.000000	0.200000	0.200000	0.200000	0.000000	0.000	0.200000	0.0	
1	0.000000	0.166667	0.166667	0.166667	0.000000	0.000	0.166667	0.0	
2	0.250000	0.000000	0.000000	0.000000	0.000000	0.250	0.250000	0.0	
3	0.111111	0.000000	0.000000	0.000000	0.111111	0.000	0.111111	0.0	

```
4  0.000000  0.000000  0.000000  0.000000  0.400000  0.000  0.200000      0.2
5  0.000000  0.000000  0.000000  0.000000  0.000000  0.375  0.250000      0.0
6  0.000000  0.000000  0.000000  0.000000  0.000000  0.000  0.500000      0.0

   meowing    orange     penny       saw     store      went
0    0.000  0.000000  0.200000  0.000000  0.000000  0.000000
1    0.000  0.166667  0.166667  0.000000  0.000000  0.000000
2    0.000  0.000000  0.000000  0.000000  0.250000  0.000000
3    0.000  0.000000  0.333333  0.111111  0.111111  0.111111
4    0.200  0.000000  0.000000  0.000000  0.000000  0.000000
5    0.125  0.125000  0.000000  0.000000  0.125000  0.000000
6    0.000  0.000000  0.500000  0.000000  0.000000  0.000000
```

Inverse Document Frequency (Inverzna dokumenta frekvencija)

Mjera koliko informacije nosi token t s obzirom na pojavnost u svim dokumentima korpusa D:

$$\text{idf}(t) = -\log \frac{\text{df}(t, D)}{|D|}$$

gdje je $\text{df}(t, D)$ broj dokumenata u D koje sadrži token t.

Paket `sklearn` omogućava korištenje FT-IDF s opcijom `use_idf=True` u `TFidfVectorizer` metodi za vektorizaciju zajedno s odabirom normalizacije vektora korištenjem L_1 ili L_2 norme.

> Napomena: `sklearn` implementira `TFidfVectorizer` idf() tako da u brojnik i nazivnik doda 1 kako bi se izbjeglo dijeljenje s 0.

[19]:
```
# iskoristimo idf
idf_vectorizer = TfidfVectorizer(stop_words='english', tokenizer=tokenizer,␣
 ↪use_idf=True, norm='l2')
X = idf_vectorizer.fit_transform(texts)
df = pd.DataFrame(X.toarray(), columns=idf_vectorizer.get_feature_names_out())
df
```

```
C:\Users\Domagoj\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.13_qbz
5n2kfra8p0\LocalCache\local-packages\Python313\site-
packages\sklearn\feature_extraction\text.py:517: UserWarning: The parameter
'token_pattern' will not be used since 'tokenizer' is not None'
  warnings.warn(
C:\Users\Domagoj\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.13_qbz
5n2kfra8p0\LocalCache\local-packages\Python313\site-
packages\sklearn\feature_extraction\text.py:402: UserWarning: Your stop_words
may be inconsistent with your preprocessing. Tokenizing the stop words generated
tokens ['ha', 'u', 'wa'] not in stop_words.
  warnings.warn(
```

[19]:
```
        ate      blue    bought    bright       bug       cat      fish  \
0  0.000000  0.512612  0.512612  0.512612  0.000000  0.000000  0.258786
1  0.000000  0.456170  0.456170  0.456170  0.000000  0.000000  0.230292
```

```
2  0.578752  0.000000  0.000000  0.000000  0.000000  0.578752  0.292176
3  0.303663  0.000000  0.000000  0.000000  0.303663  0.000000  0.153301
4  0.000000  0.000000  0.000000  0.000000  0.772313  0.000000  0.194947
5  0.000000  0.000000  0.000000  0.000000  0.000000  0.840166  0.282766
6  0.000000  0.000000  0.000000  0.000000  0.000000  0.000000  0.562463

     meowed   meowing    orange     penny       saw     store      went
0  0.000000  0.000000  0.000000  0.380417  0.000000  0.000000  0.000000
1  0.000000  0.000000  0.456170  0.338530  0.000000  0.000000  0.000000
2  0.000000  0.000000  0.000000  0.000000  0.000000  0.494698  0.000000
3  0.000000  0.000000  0.000000  0.676058  0.365821  0.259561  0.365821
4  0.465201  0.386156  0.000000  0.000000  0.000000  0.000000  0.000000
5  0.000000  0.280055  0.280055  0.000000  0.000000  0.239382  0.000000
6  0.000000  0.000000  0.000000  0.826823  0.000000  0.000000  0.000000
```

Princip klasteriranja: korištenjem k**-Means** algoritma (docs) cilj nam je odrediti k klastera dokumenata našeg korpusa tako da ukupna udaljenost između dokumenata istog klastera bude najmanja moguća!

[20]:
```
# ponovimo podatke ...
vectorizer = TfidfVectorizer(use_idf=True, tokenizer=tokenizer,␣
 ↪stop_words='english')
X = vectorizer.fit_transform(texts)
```

```
C:\Users\Domagoj\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.13_qbz
5n2kfra8p0\LocalCache\local-packages\Python313\site-
packages\sklearn\feature_extraction\text.py:517: UserWarning: The parameter
'token_pattern' will not be used since 'tokenizer' is not None'
  warnings.warn(
C:\Users\Domagoj\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.13_qbz
5n2kfra8p0\LocalCache\local-packages\Python313\site-
packages\sklearn\feature_extraction\text.py:402: UserWarning: Your stop_words
may be inconsistent with your preprocessing. Tokenizing the stop words generated
tokens ['ha', 'u', 'wa'] not in stop_words.
  warnings.warn(
```

[21]:
```
# algoritam klasteriranja
from sklearn.cluster import KMeans
# broj klastera 2
number_of_clusters = 2
km = KMeans(n_clusters=number_of_clusters)
km.fit(X)
```

[21]:
```
KMeans(n_clusters=2)
```

[22]:
```
# uljepsajmo prikaz podataka
results = pd.DataFrame()
results['text'] = texts
```

```
results['category'] = km.labels_ # klaster oznake
results
```

[22]:

	text	category
0	Penny bought bright blue fishes.	0
1	Penny bought bright blue and orange fish.	0
2	The cat ate a fish at the store.	1
3	Penny went to the store. Penny ate a bug. Penn…	1
4	It meowed once at the bug, it is still meowing…	1
5	The cat is at the fish store. The cat is orang…	0
6	Penny is a fish	1

[23]:
```
# primjer 2
from data import texts2 as texts

pprint(texts)

# inicijalizirajmo vektor s najviše 2 istaknuta svojstva: 2 najčešće riječi u␣
 ↪korpusu
vectorizer = TfidfVectorizer(use_idf=True, max_features=2,tokenizer=tokenizer,␣
 ↪stop_words='english')
X = vectorizer.fit_transform(texts)

# klasteriraj
number_of_clusters = 3
km = KMeans(n_clusters=number_of_clusters)
km.fit(X)
# citaj podatke
df = pd.DataFrame(X.toarray(), columns=vectorizer.get_feature_names_out())
df['category']=km.labels_
df
```

```
['Penny bought bright blue fishes.',
 'Penny bought bright blue and orange bowl.',
 'The cat ate a fish at the store.',
 'Penny went to the store. Penny ate a bug. Penny saw a fish.',
 'It meowed once at the bug, it is still meowing at the bug and the fish',
 'The cat is at the fish store. The cat is orange. The cat is meowing at the '
 'fish.',
 'Penny is a fish.',
 'Penny Penny she loves fishes Penny Penny is no cat.',
 'The store is closed now.',
 'How old is that tree?',
 'I do not eat fish I do not eat cats I only eat bugs']
```

```
C:\Users\Domagoj\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.13_qbz
5n2kfra8p0\LocalCache\local-packages\Python313\site-
packages\sklearn\feature_extraction\text.py:517: UserWarning: The parameter
```

```
'token_pattern' will not be used since 'tokenizer' is not None'
  warnings.warn(
C:\Users\Domagoj\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.13_qbz
5n2kfra8p0\LocalCache\local-packages\Python313\site-
packages\sklearn\feature_extraction\text.py:402: UserWarning: Your stop_words
may be inconsistent with your preprocessing. Tokenizing the stop words generated
tokens ['ha', 'u', 'wa'] not in stop_words.
  warnings.warn(
```

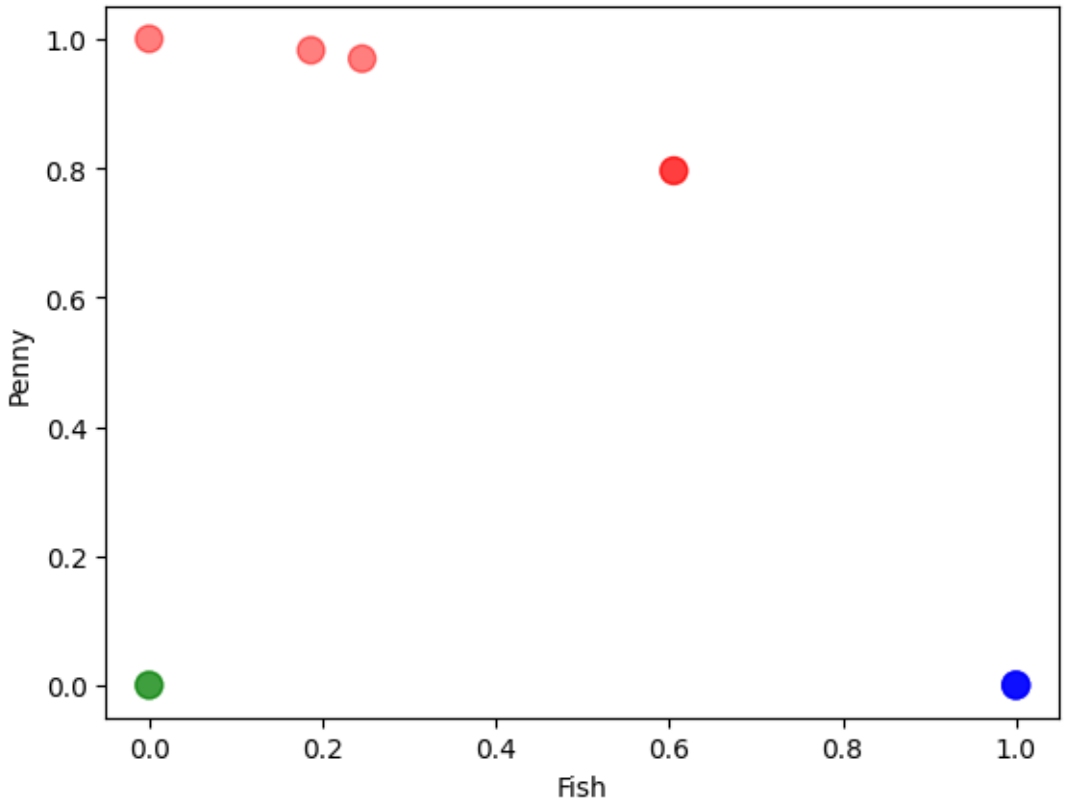
[23]:

	fish	penny	category
0	0.605349	0.795961	0
1	0.000000	1.000000	0
2	1.000000	0.000000	1
3	0.245735	0.969337	0
4	1.000000	0.000000	1
5	1.000000	0.000000	1
6	0.605349	0.795961	0
7	0.186785	0.982401	0
8	0.000000	0.000000	2
9	0.000000	0.000000	2
10	1.000000	0.000000	1

[24]:
```
# ilustracija klastera
# boje klastera
color_list = ['r', 'b','g']
colors = [color_list[i] for i in df['category']]

ax = df.plot(kind='scatter', x='fish', y='penny', alpha=0.5, s=100,c=colors)
ax.set_xlabel("Fish")
ax.set_ylabel("Penny")
```

[24]:
```
Text(0, 0.5, 'Penny')
```



- **Klasifikacija** = zadatak utvrđivanja grupe kojom pripada dani podatak.
- Postupak može biti *nenadziran* ili *nadziran*: **non-supervised/supervised**

3 Primjer: Klasifikacija teksta

Klasteriranjem podataka dobili smo **tematske kategorije** dokumenata unutar korpusa. > **Pitanje**: kako odrediti kojoj kategoriji pripada novopridošli (nekategorizirani) dokument?

Odgovor: Primijeniti K-Nearest Neighbour klasifikator (docs)!

[25]:
```
from data import texts2 as texts

# Bag-of-Words
vectorizer = TfidfVectorizer(use_idf=True, stop_words='english')
X = vectorizer.fit_transform(texts)

# klasifikator
text = 'Penny likes to fish a fish.'

from sklearn.cluster import KMeans
# broj klastera
```

```
number_of_clusters = 4
km = KMeans(n_clusters=number_of_clusters)
km.fit(X)

# ispisi rezultat
results = pd.DataFrame()
results['text'] = texts
results['category'] = km.labels_
results

from sklearn.neighbors import KNeighborsClassifier
classifier = KNeighborsClassifier(n_neighbors=3)
classifier.fit(X,km.labels_)
# test uzorak
test=vectorizer.transform([text])
#pprint(test.toarray())

# predikcija
category = classifier.predict(test)
print('Dokument je pridruzen klasteru: ', category)
```

```
Dokument je pridruzen klasteru:  [2]
```