CS 595: Assignment 10

Mallika Kogatam

Fall 2014

Contents

1 Problem 1 2
1.1 Solution 2
2 Problem 2 3
2.1 Solution 3
3 Code Listing 11
3.1 docclass.py 11
3.2 feedfilter.py 15
4 Problem 2 17
4.1 Solution 17

1 Problem 1

Question

```
Choose a blog or a newsfeed (or something similar as long as it has
an Atom or RSS feed).  It should be on a topic or topics of which you
are qualified to provide classification training data.  In other words,
choose something that you enjoy and are knowledgable of.  Find a feed
with at least 100 entries.

Create between four and eight different categories for the entries
in the feed:

examples:

work, class, family, news, deals

liberal, conservative, moderate, libertarian

sports, local, financial, national, international, entertainment

metal, electronic, ambient, folk, hip-hop, pop

Download and process the pages of the feed as per the week 12
class slides.
```

1.1 Solution

1. I am not familiar with any blogs so choosing a blog to continue with my assignment became really a big task for me.
2. I struggled a lot to find a blog which had an atom feed and the had my interest in it.
3. So I finally came up with a blog which provides information on Indian jewellery and the latest jewellery models. the website is `http://www.southjewellery.com/`
4. I came up with 4 categories named "fancy" , "celebrity" , "traditional" , "stones".
   - **Fancy** Jewellery which are designed for daily wear , and that have floral designs and stones that are curved etc.
   - **Traditional** Jewellery that have antique designs, that have indian goddess on pendants etc
   - **Celebrity** Jewellery that were worn by celebrities
   - **Stones** Jewellery that are designed by stones .
5. I downloaded 100 entries from the atom feed and saved in *jewelery.xml*

2 Problem 2

Question

```
Manually classify the first 50 entries, and then classify (using
the fisher classifier) the remaining 50 entries. Report the cprob()
values for the 50 titles as well.  From the title or entry itself,
specify the 1-, 2-, or 3-gram that you used for the string to
classify.  Do not repeat strings; you will have 50 unique strings.
For example, in these titles the string used is marked with *s:

*Rachel Goswell* - "Waves Are Universal" (LP Review)
The *Naked and Famous* - "Passive Me, Aggressive You" (LP Review)
*Negativland* - "Live at Lewis's, Norfolk VA, November 21, 1992" (concert)
Negativland - "*U2*" (LP Review)

Note how "Negativland" is not repeated as a classification string.

Create a table with the title, the string used for classification,
cprob(), predicted category, and actual category.
```

2.1 Solution

1. I used the *docclass.py* and *feedfilter.py* files found in Intelligence collective programming text book as the basis for my document filtering.
2. The modified source code, shown in Listing 2 and 1 will let me manually train the classifier using the Fisher method. Save the features and related categories to a set of database tables so that the training will persist between sessions
3. Listing 2 and 1 uses the Fisher method to predict a category based on the entry.
4. Table 1 will show the predicted and actual category for an entry along with it's summary for the first 50 entries.
5. Which means I am training the first 50 entries and storing them into various categories.
6. Basing on the first 50 trained entries, the next 50 categories were predicted by the classifier.
7. Tables 3, 4, 5, 6, 7 will show the predicted, actual, string classifier, cprob and type of string classifier for next 50 categories.

Title	summary	Predicted	Actual
Multi Layered Pacchi Floral Necklace	22 carat gold bridal multi layered floral and peacock design uncut diamond pacchi necklace with south sea pearl strings and handcrafted peacock motifs on the sides.	none	Fancy
Polki Diamond Lakshmi Chandbalis	22 carat gold antique finish Lakshmi chandbalis studded with polki diamonds, diamonds and rubies	Fancy	Stones
Floral Diamond Necklace Set	18 carat gold floral design diamond necklace and earrings from Malabar Gold and Diamonds.	Fancy	Fancy
Black Diamonds Mangalsutra Chains	22 carat gold simple black diamonds mangalsutra chains strung with south sea pearls, ruby beads, emerald beads and Nakshi balls.	Stones	stones
Uncut Diamond Necklace Set	22 carat gold bridal uncut diamond choker set studded with uncut diamonds, kundans and embellished with enamel	Fancy	Fancy
Peacock Diamond Necklace	18 carat gold peacock design diamond necklace studded with diamonds, rubies and emeralds teamed with suitable diamond earrings	Fancy	Stones
Pinky Reddy in Crystal Balls Necklace	Socialite Pinky Reddy in multi color crystal balls necklace teamed with huge emerald diamond tops.	Stones	Celebrity
Shreedevi in Diamond Necklace	Socialite Shreedevi Chowdary in MBS Jewellers diamond necklace and earrings.	Stones	Celebrity
Elegant Beads Necklace with Lakshmi Pendant	22 carat gold temple design necklace strung with two rows of emerald beads, ruby beads and south sea pearls. The necklace has a divine Goddess Lakshmi pendant studded with polki diamonds and rubies	Stones	Traditional
Nakshi Work Chandbalis	22 carat gold beautiful Nakshi work chandbalis studded with polki diamonds, rubies, emeralds and hanging south sea pearl drops	Stones	Stones
Diamond Pendant Set	18 carat gold diamond pendant and earrings studded with shimmering diamonds, kundans, rubies and south sea pearl drops.	Stones	Stones
Mugdha Art Studio Designer Sashi Engagement	Fashion designer Sashi Vangapalli of Mugdha Art Studio got engaged to Sandeep. For her engagement event, Sashi adorned a beautiful rich diamond necklace set studded with rubies; diamond haram; small tikka and gold armlets.	Stones	Celebrity
Short Black Beads Chain	22 carat gold light weight two line hort black beads mangalutra chain.	Stones	Fancy
Antique Nakshi Peacock Necklace	22 carat gold antique toned peacock design nakshi work necklace with beauitufl peacock pendant studded with polki diamonds, emeralds, rubies and hanging south sea pearl	Stones	Traditional
Traditional Lakshmi Kasu Haram	22 carat gold traditional Lakshmi kasulaperu haram studded with rubies and emeralds teamed with suitable earrings	Stones	Traditional
Light Weight Gold Balls Necklace	22 carat gold two chain light weight gold balls necklace and earrings set	Fancy	Fancy
Antique Finish Lord Vishnu Pendant	22 carat gold antique finish Lord Vishnu pendant with peacocks on the top studded with rubies and kundans	Stones	Traditional
Peacock Design Kids Vanki	22 carat gold peacock design kids vanki studded with kundans, emeralds and rubies.	Traditional	Fancy
Ruby Kundan Mango Haram	22 carat gold traditional mango mala with peacock design pendant decorated with rubies, kundans and emeralds.	Traditional	Traditional
South Sea Pearls Necklace with Divine Pendant	22 carat gold south sea pearls necklace with divine pendant studded with diamonds.	Traditional	Traditional
South Sea Pearls Flat Diamonds Necklace	22 carat gold necklace with two south sea pearl strings and a central flat diamond chain attached to floral motifs on either side studded with flat diamonds and emeralds.	Fancy	Fancy
Kundan Choker Set	22 carat gold choker set studded with kundans, flat diamonds, pearls, emerald beads and embellished with enamel.	Fancy	Fancy
Gold Haram and Jhumkas Set	22 carat gold haram studded with kundans and suitable kundan jhumkas.	Fancy	Fancy
Diamond Ruby Bangles	18 carat gold bangles studded with diamonds and rubies.	Stones	Stones
Uncut Diamond Peacock Jhumkas	22 carat gold lovely peacock design jhumkas studded with uncut diamonds, diamonds, rubies and small pearls hanging at the bottom of jhumka.	Fancy	Fancy

Table 1: Manually classified first 25 entries

Title	summary	Predicted	Actual
Peacock Design Gold Kada	22 carat gold peacock design kada pair embellished with enamel and studded with rubies and emeralds	Fancy	Fancy
Gundla Haram with Lakshmi Pendant	22 carat gold antique finish three line gundla haram with Goddess Lakshmi motifs on either side and Goddess Lakshmi pendant at its center studded with rubies and emeralds.	Traditional	Traditional
Vasundhara Jewellers Jhumkas Collection	Pretty designer jhumkas studded with diamonds and precious stones from Vasundhara Jewellers	Stones	Stones
Stunning Diamond Vaddanam	Stunning bridal diamond vaddanam studded with diamonds, rubies and emeralds from Vasundhara Jewellers.	Stones	Stones
Uncut Diamond Necklace	22 carat gold necklace studded with uncut diamonds, rubies, emeralds and polki diamonds.	Stones	Stones
Black Beads Bangles	22 carat gold black beads bangles	Traditional	Stones
Sridevi in Diamond Emerald Necklace	Actress Sridevi in a trendy south sea pearls choker with pendant studded with huge emeralds and a diamond flower motif at its center.	Traditional	Celebrity
Uncut Diamond Chandbalis	22 carat gold chandbalis studded with uncut dimaonds, rubies, emeralds and hanging cultured south sea pearls	Fancy	Fancy
Diamond Emerald Ruby Necklace	Colorful necklace and earrings set studded with diamonds, emeralds, rubies and south sea pearls. The necklace has multistring pearls chain on its back.	Traditional	Fancy
Beautiful Diamond Necklace	Beautiful diamond necklace from Malabar Gold and Diamonds studded with emerald at its center.	Fancy	Fancy
Shraddha Das in Kundan Necklace	Actress Shraddha Das in heavy kundan necklace and earrings.	Celebrity	Celebrity
Archana in Diamond Necklace Set	Actress Archana in diamond necklace floral ruby motif at its center teamed with matching diamond ruby earrings.	Fancy	Celebrity
Gold Haram with CZ Stones Pendant	22 carat gold gundla haram with beautiful peacock design stones pendant studded with cz stones, rubies and emeralds	Fancy	Fancy
Pearls and Beads Haram Models	———	Traditional	Traditional
Traditional Ruby Choker and Jhumkas	Socialite in antique finish traditional ruby choker and jhumkas.	Traditional	Celebrity
Gold Haram with Lakshmi Pendant	22 carat gold beautiful gold haram studded with rubies, polki diamonds and emeralds. The necklace has an inner row of south sea pearls string and Goddess Lakshmi pendant.	Traditional	Traditional
Gorgeous Diamond Necklace and Jhumkas	Stunning diamond necklace sprinkled with shimmering diamonds, emeralds and south sea pearls teamed with attractive diamond emerald jhumkas from Malabar Gold and Diamonds.	Fancy	Fancy
Simple Emerald Necklace	22 carat gold simple emerald necklace studded with rubies and cz stones.	Fancy	Fancy
Nikitha Reddy in Chandbalis	Producer Nikitha Reddy in gold necklace studded with rubies and pearls teamed with gold chandbalis.	Traditional	Celebrity
Beautiful CZ Stones Vaddanam and Bajubandh Set	22 carat gold peacock design beautiful vaddanam and bajubandh set studded with cz stones, emeralds and rubies.	Fancy	Fancy
Polki Diamond Bali Hoop Earrings	Antique finish 22 carat gold bali hoop earrings with elephant design studded with polki diamonds and emeralds and small pearls hanging at the bottom.	Traditional	Traditional
Bride in Diamond Jewellery	South Indian bride in her wedding jewellery. She adorned a pearls choker with diamond pendant; medium length diamond haram; long diamond mango mala and diamond vaddanam.	Celebrity	Celebrity.

Table 2: Manually classified next 25 entries from 25 to 50 entries

Title	summary	Predicted	Actual	String	Cprob	Type
Nakshi Balls Haram	22 carat gold antique finish broad choker and nakshi balls haram studded with kundans from Malabar Gold and Diamonds.	Traditional	Traditional	Haram	0.0	1-gram
Traditional Mango Haram	22 carat gold mango haram with Goddess Lakshmi pendant studded with rubies and emeralds.	Fancy	Traditional	Mango haram	0.0	2-gram
Celebs in Heavy Chandbalis and Jhumkas	Bollywood celebs wore heavy jhumkas and chandbalis without adding any neck jewellery for the wedding reception of Arpita Khan and Aaysuh Sharma.	Fancy	Celebrity	Arpita	0.0	1-gram
22 Carat Gold Long Jhumkas	22 carat gold antique finish gold long jhumkas studded with kundans from Shree Raj Mahal Jewellers.	Stones	Traditional	Long jhumkas	0.0	2-gram
Latest Temple Jewellery Designs	22 carat gold antique finish latest temple jewellery designs from Malabar Gold and Diamonds. Featuring a short peacock necklace; medium length temple haram	Traditional	Traditional	Temple Jewellery	0.0	2-gram
Ruby Diamond Necklace	Bridal diamond necklace studded with large rubies and small diamonds from Malabar Gold and Diamonds.	Fancy	Stones	Large rubies	0.0	2-gram
Malabar Gold Bridal Jewellery	Broad uncut diamond necklace, uncut diamond layered long haram and nakshi work temple design vaddanam studded with precious stones from Malabar Gold and Diamonds.	Fancy	Traditional	Nakshi work	0.0	2-gram
Layered Polki and Uncut Diamond Necklace	Attractive bridal necklace from Malabar Gold and Diamonds featuring a central ruby and uncut diamond motifs row with south sea pearl and polki diamond chains on either side and tear drop shaped polki diamond pendant. The necklace is complemented by matching polki diamond earrings.	Fancy	Fancy	Polki diamond	0.0	2-gram
Broad Uncut Diamond Necklace Set	22 carat gold bridal uncut diamond necklace from Malabar Gold and Diamonds studded with uncut chakri diamonds and rubies in intricate design teamed with suitable uncut diamond jhumkas.	Fancy	Stones	Uncut diamond	0.0	2-gram
3 in 1 Diamond Haram cum Vaddanam cum Pendant	18 carat gold 3 in 1 paisley design diamond haram with peacock design pendant.This diamond haram can be worn as vaddanam or pendant set. It is teamed with matching diamond jhumkas.	Fancy	Fancy	3 in 1	0.0	3-gram

Table 3: Automatically classified 10 entries of next 50 entries

Title	summary	Predicted	Actual	String	Cprob	Type
Pretty Diamond Necklace	18 carat gold beautiful diamond necklace studded with diamonds, rubies and south sea pearl drops from Malabar Gold and Diamonds.	Stones	Stones	Diamond necklace	0.0	2-gram
Beads Mala with Diamond Mugappu	22 carat gold trendy beads mala string with south sea pearls and ruby beads and a diamond mugappu/side pendant studded with diamonds and emerald stone at its center from Malabar Gold and Diamonds.	Traditional	Fancy	Trendy	0.0	1-gram
Chakri Uncut Diamond Necklace and Jhumkas	22 carat gold chakri uncut diamond necklace with round pendant paired with matching long uncut diamond jhumkas from Malabar Gold and Diamonds.	Fancy	Fancy	Chakri uncut diamond	0.0	3-gram
Polki Diamond Necklace and Jhumkas	Gorgeous bridal polki diamond necklace from Malabar Gold and Diamonds studded with several rows of rich polkis, emeralds and ruby. The necklace is complemented by beautiful polki jhumkas.	Fancy	Stones	Polkis, emeralds and ruby	0.0	3-gram
Heavy Chandbali Style Jhumkas	Socialite looking gorgeous in huge chandbali style jhumkas and gold bib style antique finish necklace.	Celebrity	Celebrity	Socialite looking gorgeous	0.0	3-gram
Colorful Navaratna Haram	Antique finish gold navaratna haram studded with nine types of gemstones and a central two step pendant studded with emeralds and diamonds.	Traditional	Traditional	Navaratna	0.0	1-gram
Gundla Mala and Kundan Choker	Socialite in two line gundla mala with diamond pendant; kundan choker and antique finish gold jhumkas.	Celebrity	Celebrity	Socialite	0.0	1-gram
Heavy Antique Pendant and Jhumkas	Socialite looking elegant in heavy antique temple pendant, antique long jhumkas and precious stones studded vaddanam.Checkout Nara Brahmini in the same necklace at a recent wedding event!!	Traditional	Celebrity	Nara brahmini	0.0	2-gram
2 in 1 Haram cum Baby Waist Belt	22 carat gold antique finish gold balls haram with Goddess Lakshmi pendant studded with rubies and emeralds. This gundla mala can also be used as baby waist belt.	Traditional	Traditional	Goddess	0.5	1-gram
Ruby Emerald Bangles	22 carat gold bangles studded with rubies, emeralds and white stones.	Stones	Stones	Studded with rubies	0.0	3-gram

Table 4: Automatically classified next 10 entries from 60 to 70 entries

Title	summary	Predicted	Actual	String	Cprob	Type
Meena in Gold Necklace	Actress Meena attended Pandiarajan son's wedding reception. For the occasion she adorned a broad gold necklace and simple gold jhumkas.	Fancy	Celebrity	Meena	0.0	1-gram
Bridal Diamond Necklace	Gorgeous bridal diamond necklace with intricate design studded with shimmering diamonds, emeralds and south sea pearl drops from Shree Raj Mahal Jewellers	Stones Stones	Shimmering diamonds	0.0	2-gram	
Actor Pandiarajan Son Wedding Reception	Actor Pandiarajans eldest son Pallavarajans wedding reception was held on Nov 20 in Chennai.The bride wore a diamond necklace, matching earrings, tikka and armlet studded with diamonds and rubies.	Fancy	Celebrity	Actor Pandiarajan	0.0	2-gram
Sangeetha in Diamond Necklace and Jhumkas	Sangeetha attended actor Pandiarajan son's wedding reception. For the occassion, Sangeetha adorned a diamond necklace and three step diamond jhumkas	Traditional	Celebrity	Sangeetha	0.0	1-gram
Beautiful Polki Chandbalis	Beautiful latest model chandbali earrings studded with polki diamonds, rubies, emeralds and south sea pearl drops.	Stones	Stones	South sea pearl	0.0	3-gram
Peacock Mango Pacchi Necklace	22 carat gold antique finish gorgeous peacock mango pacchi necklace with intricate Nakshi work studded with uncut polki diamonds, emeralds, rubies and south sea pearl drops	Stones	Fancy	Peacock mango pacchi	0.0	3-gram
22 Carat Gold Tussi Necklace	22 carat gold traditional tussi necklace studded with polkis, teamed up with suitable jhumkas	Traditional	Traditional	Tussi	0.0	1-gram
Arpita Khans Wedding Jewellery	For both mehendi ceremony and wedding event, Arpita Khan adorned heavy kundan jewellery.	Celebrity	Celebrity	Arpita	0.0	1-gram
Simple Black Beads Mangalsutra	22 carat gold light weight black beads mangalsutra chain	Fancy	Fancy	Light	0.0	1-gram
Trisha in NAC Antique Jewellery	Actress Trisha Krishnan is looking gorgeous in NAC antique necklace featuring Goddess Lakshmi pendant studded with polki diamonds; matching chandbali earrings and broad gold kada studded with emeralds and diamonds.	Traditional	Celebrity	Trisha	0.0	1-gram

Table 5: Automatically classified next 10 entries from 70 to 80 entries

Title	summary	Predicted	Actual	String	Cprob	Type
Diamond Peacock Earrings	Lovely peacock earrings studded with diamonds, kundans and pearls and embellished with enamel work.	Fancy	Fancy	embellished with enamel	0.0	3-gram
Pacchi Mango Necklace	22 carat gold pacchi mango necklace studded with polki diamonds, diamonds, rubies and emerald from Mor Jewellers.	Stones	Stones	Rubies and emerald	0.0	3-gram
Pretty Diamond Chandbalis	Cute diamond chandbali earrings in medium size studded with diamonds, rubies, emeralds and south sea pearl drops	Stones	Stones	Diamond chandbali	0.0	2-gram
5 Grams Simple Black Beads Mangalsutra	22 carat gold light weight simple black beads mangalsutra chain strung with gold balls and south sea pearls	Stones	Fancy	5 grams	0.0	1-gram
Ruby Mango Mala	22 carat gold beautiful traditional mango mala studded with rubies, emeralds and uncut diamonds	Fancy	Traditional	Mala	0.0	1-gram
Khushbu Sundar in Gundla Mala	Actress Khushbu Sundar in two line gold gundla mala strung with a few turquoise beads.	Traditional	Celebrity	khushbu	0.0	1-gram
Light Weight Uncut Diamond Chandbalis	22 carat gold light weight uncut diamond chandbalis with south sea pearl drops from Kothari Jewellers.	Stones	Stones	uncut Diamond	0.0	2-gram
Pacchi Mango Necklace	Beautiful pacchi mango necklace studded with polki diamonds, rubies, emeralds and south sea pearl drops and finished with intricate Nakshi work.	Traditional	Traditional	Pacchi	0.0	1-gram
Bridal Diamond Necklace	Dazzling diamond necklace sprinkled with shimmering diamonds and a large emerald from Shree Rajmahal Jewellers.	Stones	Stones	bridal diamond	0.0	2-gram
Simple Diamond Bangles	18 carat gold simple and elegant diamond bangles from Vummidi Jewellers.	Stones	Stones	Elegant diamond	0.0	2-gram

Table 6: Automatically classified next 10 entries form 80 to 90 entries

Title	summary	Predicted	Actual	String	Cprob	Type
Elegant Pearls Mala with Carved Ruby Pendant	Eye-catching south sea pearls strand with carved ruby pendant topped with uncut diamond flower motif from Umrao Jewels.	Traditional	Fancy	Carved ruby	0.0	2-gram
Anushka in Two Tier Jhumkas	Ansuhka in two tier gold jhumkas studded with rubies and turquoise stones.	Fancy	Celebrity	Anushka	0.0	1-gram
Broad Gold	Kada Broad gold kada crafted in 22 carat gold with intricate design from Vummidi Jewellers.	Traditional	Fancy	Crafted	0.0	1-gram
Multistring Pearls Necklace with Pendant	Gorgeous multistring pearls necklace with beautiful gold pendant studded with rubies, kundans and emerald from Shree Raj Mahal Jewellers.	Stones	Fancy	Multistring	0.0	1-gram
Dazzling Ruby Diamond Earrings	Latest ruby diamond earrings in 18 carat gold that can be teamed up with any outfit.	Stones	Fancy	Dazzling	0.0	1-gram
Exquisite Polki Diamond Waist Belt	22 carat gold stunning temple Nakshi work vaddanam studded with polki diamonds in pacchi setting, rubies and emeralds	Stones	Traditional	Temple nakshi	0.0	2-gram
Sukumar Wife Tabitha in Mango Necklace	Tollywood director Sukumar wife Tabitha looked elegant in traditional saree teamed up with ruby and cz stones mango necklace set at "Chakkiligintha" audio release function.	Celebrity	Celebrity	sukumar	0.0	1-gram
Nakshi Work Mango Peacock Necklace	22 carat gold antique finish gorgeous mango peacock design necklace with intricate Nakshi work studded with polki diamonds in pacchi setting, emerald and rubies	Traditional	Traditional	Intricate nakshi work	0.0	3-gram
Lord Krishna Antique Jhumkas	22 carat gold elegant antique finish Nakshi work jhumkas with Lord Krishna tops studded with uncut diamonds, emerald beads and pearls.	Traditional	Traditional	lord krishna antique	0.0	3-gram
Antique Kundan Earrings	22 carat gold antique finish lovely pair of earrings studded with kundans and south sea pearl drops.	Traditional	Traditional	Antique kundan	0.0	2-gram

Table 7: Automatically classified next 10 entries from 90 to 100

3 Code Listing

3.1 docclass.py

```
#from pysqlite2 import dbapi2 as sqlite
from sqlite3 import dbapi2 as sqlite
import re
import math

def getwords(doc):
  splitter=re.compile('\\W*')
  #print doc
  ## Remove all the HTML tags
  doc=re.compile(r'<[^>]+>').sub('',doc)
  # Split the words by non-alpha characters
  words=[s.lower() for s in splitter.split(doc)
         if len(s)>2 and len(s)<20]

  # Return the unique set of words only
  return dict([(w,1) for w in words])

class classifier:
  def __init__(self,getfeatures,filename=None):
    # Counts of feature/category combinations
    self.fc={}
    # Counts of documents in each category
    self.cc={}
    ## extract features for classification
    self.getfeatures=getfeatures

  def setdb(self,dbfile):
    self.con=sqlite.connect(dbfile)
    self.con.execute('create table if not exists rss(num, entry, feature, predicted, actual,
        cprob)')
    self.con.execute('create table if not exists fc(feature,category,count)')
    self.con.execute('create table if not exists cc(category,count)')
    # remove old data from previous sessions
    self.con.execute('delete from rss')
    self.con.execute('delete from fc')
    self.con.execute('delete from cc')

  def manualClassdb (self,num, entry, feature, predicted, actual):
    self.con.execute("insert into rss values ('%s','%s', '%s', '%s','%s', '%s')"
                     % (num, entry, feature, predicted, actual, None))
    self.con.commit()

  def autoClassdb (self,num, entry, feature, predicted, actual, cp):
    self.con.execute("insert into rss values ('%s','%s', '%s', '%s','%s', '%s')"
                     % (num, entry, feature, predicted, actual, cp))
    self.con.commit()
  ## Increase the count of a feature/category pair
  def incf(self,f,cat):
    count=self.fcount(f,cat)
    if count==0:
      self.con.execute("insert into fc values ('%s','%s',1)"
                       % (f,cat.lower()))
    else:
      self.con.execute(
        "update fc set count=%d where feature='%s' and category='%s'"
        % (count+1,f,cat.lower()))

  ## The number of times a feature has appeared in a category
  def fcount(self,f,cat):
    res=self.con.execute(
      'select count from fc where feature="%s" and category="%s"'
      %(f,cat)).fetchone()
    if res==None: return 0
    else: return float(res[0])

  ## Increase the count of a category
  def incc(self,cat):
```

```
67      count=self.catcount(cat)
68      if count==0:
69        self.con.execute("insert into cc values ('%s',1)" % (cat.lower()))
70      else:
71        self.con.execute("update cc set count=%d where category='%s'"
72                         % (count+1,cat))
73
74    ## The number of items in a category
75    def catcount(self,cat):
76      res=self.con.execute('select count from cc where category="%s"'
77                           %(cat)).fetchone()
78      if res==None: return 0
79      else: return float(res[0])
80
81    ## The list of all categories
82    def categories(self):
83      cur=self.con.execute('select category from cc');
84      return [d[0] for d in cur]
85
86    ## The total number of items
87    def totalcount(self):
88      res=self.con.execute('select sum(count) from cc').fetchone();
89      if res==None: return 0
90      return res[0]
91
92
93    ## The train method takes an item(document) and a classification.
94    ## It uses the getfeatures function to the break the item into its
95    ## separate features. It then calls incf to increase the counts for
96    ## this classification for every feature. Finally, it increases
97    ## the total count for this classification.
98    def train(self,item,cat):
99      features=self.getfeatures(item)
100     # Increment the count for every feature with this category
101     for f in features:
102       self.incf(f,cat)
103
104     # Increment the count for this category
105     self.incc(cat)
106     self.con.commit()
107
108   ## Probability is a number between 0 and 1, indicating
109   ## the likelihood of an event. You calculate the probability of
110   ## a word in a particular category by dividing the number of
111   ## times the word appears in a document in that category
112   ## by the total number of documents in the category.
113   def fprob(self,f,cat):
114     if self.catcount(cat)==0: return 0
115
116     # The total number of times this feature appeared in this
117     # category divided by the total number of items in this category
118     return self.fcount(f,cat)/self.catcount(cat)
119
120   def weightedprob(self,f,cat,prf,weight=1.0,ap=0.5):
121     # Calculate current probability
122     basicprob=prf(f,cat)
123
124     # Count the number of times this feature has appeared in
125     # all categories
126     totals=sum([self.fcount(f,c) for c in self.categories()])
127
128     # Calculate the weighted average
129     bp=((weight*ap)+(totals*basicprob))/(weight+totals)
130     return bp
131
132
133
134
135 class naivebayes(classifier):
136
137   def __init__(self,getfeatures):
138     classifier.__init__(self,getfeatures)
```

```
139     self.thresholds={}
140
141   def docprob(self,item,cat):
142     features=self.getfeatures(item)
143
144     # Multiply the probabilities of all the features together
145     p=1
146     for f in features: p*=self.weightedprob(f,cat,self.fprob)
147     return p
148
149   def prob(self,item,cat):
150     catprob=self.catcount(cat)/self.totalcount()
151     docprob=self.docprob(item,cat)
152     return docprob*catprob
153
154   def setthreshold(self,cat,t):
155     self.thresholds[cat]=t
156
157   def getthreshold(self,cat):
158     if cat not in self.thresholds: return 1.0
159     return self.thresholds[cat]
160
161   def classify(self,item,default=None):
162     probs={}
163     # Find the category with the highest probability
164     max=0.0
165     for cat in self.categories():
166       probs[cat]=self.prob(item,cat)
167       if probs[cat]>max:
168         max=probs[cat]
169         best=cat
170
171     # Make sure the probability exceeds threshold*next best
172     for cat in probs:
173       if cat==best: continue
174       if probs[cat]*self.getthreshold(best)>probs[best]: return default
175     return best
176
177 ## This function will return the probability that an item with the
178 ## specified feature belongs in the specified category, assuming there
179 ## will be an equal number of items in each category.
180 class fisherclassifier(classifier):
181   def cprob(self,f,cat):
182     # The frequency of this feature in this category
183     clf=self.fprob(f,cat)
184     if clf==0: return 0
185
186     # The frequency of this feature in all the categories
187     freqsum=sum([self.fprob(f,c) for c in self.categories()])
188
189     # The probability is the frequency in this category divided by
190     # the overall frequency
191     p=clf/(freqsum)
192
193     return p
194
195
196   def fisherprob(self,item,cat):
197     # Multiply all the probabilities together
198     p=1
199     features=self.getfeatures(item)
200     for f in features:
201       p*=(self.weightedprob(f,cat,self.cprob))
202
203     # Take the natural log and multiply by -2
204     fscore=-2*math.log(p)
205
206     # Use the inverse chi2 function to get a probability
207     return self.invchi2(fscore,len(features)*2)
208
209   ## Inverse chi-squared function
210   def invchi2(self,chi, df):
```

```
211     m = chi / 2.0
212     sum = term = math.exp(-m)
213     for i in range(1, df//2):
214         term *= m / i
215         sum += term
216     return min(sum, 1.0)
217
218   def __init__(self,getfeatures):
219     classifier.__init__(self,getfeatures)
220     self.minimums={}
221
222   def setminimum(self,cat,min):
223     self.minimums[cat]=min
224
225   def getminimum(self,cat):
226     if cat not in self.minimums: return 0
227     return self.minimums[cat]
228
229   def classify(self,item,default=None):
230     # Loop through looking for the best result
231     best=default
232     max=0.0
233     for c in self.categories():
234       p=self.fisherprob(item,c)
235       # Make sure it exceeds its minimum
236       if p>self.getminimum(c) and p>max:
237         best=c
238         max=p
239     return best
```

Listing 1: docclass.py

3.2 feedfilter.py

```
1
2  import feedparser
3  import re
4  import math
5
6  import docclass.docclass as docclass
7  # Takes a filename or URL of a blog feed and classifies the entries
8  def read(feed,classifier):
9
10   splitRegexp = re.compile( r"<[^>]+>" )
11
12   num=0
13   # Get feed entries and loop over them
14   f=feedparser.parse(feed)
15   print
16   print '----- Begin manual classification (training) -----'
17   for entry in f['entries'][0:50]:
18     num=num +1
19     # Print the contents of the entry
20     title=entry['title'].encode('utf-8').replace("'","")
21     print 'Title:      '+ title
22
23     summary = splitRegexp.sub( "", entry[ "summary" ] )
24
25     print summary #entry['summary'].encode('utf-8')
26
27     # Combine all the text to create one item for the classifier
28     #fulltext='%s\n%s\n%s' % (entry['title'],entry['publisher'],entry['summary'])
29     fulltext='%s\n%s' % (entry['title'],entry['summary'])
30     # Remove apostrophes
31     fulltext=fulltext.replace("'","")
32     # Print the best guess at the current category
33     predicted=str(classifier.classify(fulltext))
34     print 'Predicted category: ', predicted
35
36     # Ask the user to specify the correct category and train on that
37     actual=raw_input('Actual category: ')
38     feature=None
39     classifier.train(fulltext, actual)
40
41     # Save the manual classifications
42     # num, entry, feature, predicted, actual, cprob=None
43     classifier.manualClassdb(num, title, feature, predicted, actual)
44
45 #def autoClassify(feed,classifier):
46   num=50
47   print '----- Begin automatic classification -----'
48   # Get feed entries and loop over them
49   #f=feedparser.parse(feed)
50   for entry in f['entries'][50:100]:
51     num=num+1
52     # Print the contents of the entry
53     title=entry['title'].encode('utf-8').replace("'","")
54     print 'Title:      '+ title
55     summary = splitRegexp.sub( "", entry[ "summary" ] )
56
57     print summary #entry['summary'].encode('utf-8')
58
59     # Combine all the text to create one item for the classifier
60     #fulltext='%s\n%s\n%s' % (entry['title'],entry['publisher'],entry['summary'])
61     fulltext='%s\n%s' % (entry['title'],entry['summary'])
62     fulltext=fulltext.replace("'","")
63     # Print the best guess at the current category
64     predicted=str(classifier.classify(fulltext))
65     print 'Predicted: ', predicted
66
67     # Ask the user to specify the correct category
68     #actual=raw_input('Enter actual category: ')
69     feature=raw_input('Enter string classifier: ')
70
```

```
71      #classifier.train(entry,cl)
72      # probability the item should be in this category
73      cp=round(classifier.cprob(feature,predicted),3)
74      print 'cprob: ', str(cp)
75      # Save the trained classifications
76      # num, entry, feature, predicted, actual, cprob(feature, predicted)
77      classifier.autoClassdb(num, title, feature, predicted, actual, cp)
78    #return classifier
79
80  def entryfeatures(entry):
81    splitter=re.compile('\\W*')
82    f={}
83
84    # Extract the title words and annotate
85    titlewords=[s.lower() for s in splitter.split(entry['title'])
86            if len(s)>2 and len(s)<20]
87    for w in titlewords: f['Title:'+w]=1
88
89    # Extract the summary words
90    summarywords=[s.lower() for s in splitter.split(entry['summary'])
91            if len(s)>2 and len(s)<20]
92
93    # Count uppercase words
94    uc=0
95    for i in range(len(summarywords)):
96      w=summarywords[i]
97      f[w]=1
98      if w.isupper(): uc+=1
99
100     # Get word pairs in summary as features
101     if i<len(summarywords)-1:
102       twowords=' '.join(summarywords[i:i+1])
103       f[twowords]=1
104
105   # Removed: Keep creator and publisher whole
106   #f['Publisher:'+entry['publisher']]=1
107
108   # UPPERCASE is a virtual word flagging too much shouting
109    if float(uc)/len(summarywords)>0.3: f['UPPERCASE']=1
110
111    return f
112
113 def main():
114    cl=docclass.fisherclassifier(docclass.getwords)
115    cl.setdb('mallika-jewelery.db')
116    read('jewelery.xml',cl)
117
118 if __name__ == "__main__":
119    main()
```

Listing 2: Python code for classifying entries

4 Problem 2

Question

```
Assess the performance of your classifier in each of your categories
by computing precision, recall, and F1.  Note that the definitions
of precisions and recall are slightly different in the context of
classification; see:

http://en.wikipedia.org/wiki/Precision_and_recall#Definition_.28classification_context.29

and

http://en.wikipedia.org/wiki/F1_score
```

4.1 Solution

1. In order to calculate the Precision , recall and F1 values TP, TN, FP, FN values are needed.
2. So I calculate the TP, TN, FP, FN values for each category. The table 8 shows these values for each category.
3. The Precision , Recall , F1 are calculated based on the following

$$Precesion = \frac{TP}{TP + FP}$$
$$Recall = \frac{TP}{TP + FN}$$
$$F1 = \frac{2TP}{2TP + FP + FN}$$

4. Calculated Precision, Recall and F1 values are shown in the table 9.

Category	**TN**	**TP**	**FN**	**FP**
Celebrity	38	4	8	0
Fancy	28	6	7	9
Stones	32	9	3	6
Traditional	29	9	5	7

Table 8: TN, TP, FN, FP values for each category

Category	**Precision**	**Recall**	**F1**
Celebrity	1	0.333	0.5
Fancy	0.4	0.462	0.428
Stones	0.6	0.75	0.66
Traditional	0.563	0.643	0.6

Table 9: Precision, Recall and F1 values for each category

Bibliography

[1] Precision and recall. http://en.wikipedia.org/wiki/Precision_and_recall#Definition_.28classification_context.29.

[2] Toby Segaran. *Programming Collective Intelligence*. O'Reilly, O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472, first edition, August 2007.