**ML + SL - HOW WE USED MACHINE LEARNING TO IDENTIFY WHAT SORT OF TECHNOLOGISTS**

**Article link** : <https://medium.com/@dhanushgopinath/how-we-used-machine-learning-to-identify-what-sort-of-technologists-register-on-geektrust-bdcbc59332db?lipi=urn%3Ali%3Apage%3Ad_flagship3_company%3B%2F4tt1U7WSuqYvCqnVEzyAQ%3D%3D>

**Machine Learning and Supervised Learning usage,**

* [Geektrust](https://geektrust.in/) Company use Supervised Learning for match the user profiles for the requirements of the companies.
* In addition they matched the best developers who had a close similarity to what the company was looking for.
* They took small steps into the world of Machine Learning at that time by doing similarity matches.

#### Training Data

Use **supervised learning**, which means classifying data based on previously labeled training data.

The first thing was to list down the features we wanted to identify in a developer to classify him/her as a Backend or Frontend or Full stack developer. Finally take a count of each column and make it as a table format or text.

Example:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Backendcount** | **Frontendcount** | **Fullstackcount** | **Mobiledevcount** | **Devopscount** | **Qacount** | **Productcount** | **Role** |
| 4 | 0 | 2 | 2 | 0 | 0 | 0 | 0 |
| 4 | 6 | 10 | 1 | 0 | 0 | 0 | 2 |
| 4 | 1 | 4 | 2 | 0 | 0 | 0 | 2 |
| 1 | 0 | 1 | 2 | 0 | 0 | 0 | 2 |
| 6 | 1 | 5 | 1 | 0 | 0 | 0 | 2 |
| 9 | 1 | 4 | 4 | 0 | 0 | 0 | 2 |

#### Training & Testing

Once the training data was ready we had to test the accuracy of this model. We tried different classifiers like,

* K Nearest Neighbours (KNN),
* Decision Trees ,
* Support Vector Machines (SVM).

They use **Python** and **SVM** method to classify their profile, every profile update the developer does on Geektrust, they build the feature data and predict the role from it.

**Demo – Python resume parser**

**Sample Application Using Supervised Learning (SL)**

**Note:** Save the entire files in same folder.

Step 1:

Install Python (I am using python3)

Open command prompt -> Execute following command to install PyWin32 lib

**“pip install pypiwin32”**

Step 2:

Prepare resumes to classify based on keywords

Step 3:

Open new file in notepad++ - > copy the following code and paste it -> save it as **resume\_identifier.py**

# **resume\_identifier.py**

from time import sleep

import win32com.client as win32

import glob,os

import sys

#Make sure that this file is in the same directory as the files that are to be identified as Software or Management resume.

Files = []

extensions = ['\*.docx','\*.doc','\*.rtf']

software = []

management = []

m\_map = {}

s\_map = {}

print ('Are the resume that are to be checked in this directory: '+os.getcwd()+' (Y/N)')

if str(input(':')) == 'N':

print ('Please paste resume\_identifier.py in the same directory as the word documents that are to be identified and Try again.')

print ('This program is being terminated.')

sleep(5)

sys.exit()

word = win32.gencache.EnsureDispatch('Word.Application')

word.Visible = False

for e in extensions:

for infile in glob.glob( os.path.join('',e) ):

Files.append(infile)

if ('Management keyterms.docx' in Files) & ('Software keyterms.docx' in Files):

for infile in ['Management keyterms.docx','Software keyterms.docx']:

m = {}

doc = word.Documents.Open(os.getcwd()+'\\'+infile)

for each\_word in doc.Words:

w = ""

text = each\_word.Text

for i in text:

if ((i>='a') & (i<='z')) |((i>='A') & (i<='Z')) | (i == '-') :

w+=i

else:

break

if w!="":

m[w] = 1

if infile == 'Software keyterms.docx':

s\_map = m

else:

m\_map = m

print ("Done Parsing ",infile)

doc.Close(False)

sleep(1)

for infile in Files:

m = {}

if '~' in infile:

os.remove(infile)

continue

if infile in ['Management keyterms.docx','Software keyterms.docx']:

continue

else:

print ('Do you wish to identify '+infile+' resume? (Y/N) ')

choice = str(input(':'))

if choice == 'N':

continue

elif choice != 'Y':

print ('Invalid input')

print (infile + ' is being identified')

scount = 0

mcount = 0

doc = word.Documents.Open(os.getcwd()+'\\'+infile)

for each\_word in doc.Words:

w = ""

text = each\_word.Text

for i in text:

if ((i>='a') & (i<='z')) |((i>='A') & (i<='Z')) | (i == '-') :

w+=i

else:

break

if w!="":

if w in m.keys():

m[w] += 1

else:

m[w] = 1

if w in s\_map.keys():

#print 'software',w

scount += 1

if w in m\_map.keys():

#print 'management',w

mcount += 1

print ("Done Parsing ",infile)

doc.Close(False)

sleep(1)

spercent = int(float(scount)/float(len(s\_map.keys()))\*100)

mpercent = int(float(mcount)/float(len(m\_map.keys()))\*100)

print (mcount,' words from ',infile,' matches out of ',len(m\_map.keys()),' management keywords which is ',mpercent,'%')

print (scount,' words from ',infile,' matches out of ',len(s\_map.keys()),' software keywords which is ',spercent,'%')

if spercent > mpercent:

software.append(infile)

else:

management.append(infile)

else:

print ('Software keyterms.docx Or Management keyterms.docx Not Found')

print ('Execute keyword\_generator.py before executing this program')

print ('This program is being terminated.')

sleep(5)

sys.exit()

word.Application.Quit(-1)

print ('Resume identified as software resume using "software keyterms.docx" are:')

for i in software:

print (i)

print ('Resume identified as management resume using "management keyterms.docx" are: ')

for i in management:

print (i)

sleep(10)

Step 4:

Open new file in notepad++ - > copy the following code and paste it -> save it as **keywords\_generator.py (**Flowing code only for Management/Software resume keyword generator**)**

**#keyword\_generator.py**

from time import sleep

import win32com.client as win32

import glob,os

Files = []

map\_list = []

extensions = ['\*.docx','\*.doc','\*.rtf']

#software = ['Shailender Dabodiya\_BA Continuum\_5.06\_yrs.docx','Manisha Bhayana elhi 8.03 rs.doc']

#management = ['Nitin Bailey\_Sales Manager -- India.docx','Philip Sales.docx','Matiur\_rahiman786@yahoo.com.doc','PAWAN KUMAR KANDPAL \_Admin.doc','SAMEER LEEKHA\_ CTO.rtf']

software = []

management = []

m\_maps = []

s\_maps = []

word = win32.gencache.EnsureDispatch('Word.Application')

word.Visible = False

for e in extensions:

for infile in glob.glob( os.path.join('',e) ):

if '~' in infile:

#make sure that word doesn't contain ~ symbols.

#This is to avoid the script from picking upfiles that are being

#edited which give an exception of improper word document

continue

if infile in ['Management keyterms.docx','Software keyterms.docx']:

#to remove the already existing key terms word documents if existing

os.remove(infile)

else:

sleep(1)

m = {}

Files.append(infile)

print ('Do you want to consider '+infile+' for generating the key terms? (Y/N) ')

choice = str(input(':'))

if choice == 'N':

continue

elif choice != 'Y':

print ('Invalid input')

print (infile + 'is being used for generating key terms')

print ('Is '+infile+' a management resume or a software resume? (M/S)')

category = str(input(':'))

if category == 'S':

software.append(infile)

elif category == 'M':

management.append(infile)

else:

print ('Invalid input')

print (infile + 'is considered to be a software resume')

software.append(infile)

doc = word.Documents.Open(os.getcwd()+'\\'+infile)

for each\_word in doc.Words:

w = ""

text = each\_word.Text

for i in text:

if ((i>='a') & (i<='z')) |((i>='A') & (i<='Z')) | (i == '-') :

w+=i

else:

break

if w!="":

if w in m.keys():

m[w] += 1

else:

m[w] = 1

if infile in software:

s\_maps.append(m)

elif infile in management:

m\_maps.append(m)

else:

map\_list.append(m)

print ("Done Parsing ",infile)

doc.Close(False)

sleep(1)

m\_common = m\_maps[0]

doc = word.Documents.Add()

sleep(1)

pointer = doc.Range(0,0)

for i in range(1,len(m\_maps)):

temp\_map = m\_maps[i]

for k in m\_common.keys():

if k in temp\_map:

continue

else:

del m\_common[k]

for j in m\_common.keys():

pointer.InsertAfter(j + '\n')

#print "common management terms are: "

#for i in m\_common.keys():

# print i

doc.SaveAs(os.getcwd()+'\\Management keyterms.docx')

print ('Management common terms word document created in the present directory')

doc.Close(False)

s\_common = s\_maps[0]

doc = word.Documents.Add()

sleep(1)

pointer = doc.Range(0,0)

for i in range(1,len(s\_maps)):

temp\_map = s\_maps[i]

for k in s\_common.keys():

if k in temp\_map:

continue

else:

del s\_common[k]

for j in s\_common.keys():

pointer.InsertAfter(j + '\n')

doc.SaveAs(os.getcwd()+'\\Software keyterms.docx')

print ('Software common terms word document created in the present directory')

doc.Close(False)

word.Application.Quit(-1)

print ('Key words generated')

print ('Now the script for the identification of resume can be used')

Step 5:

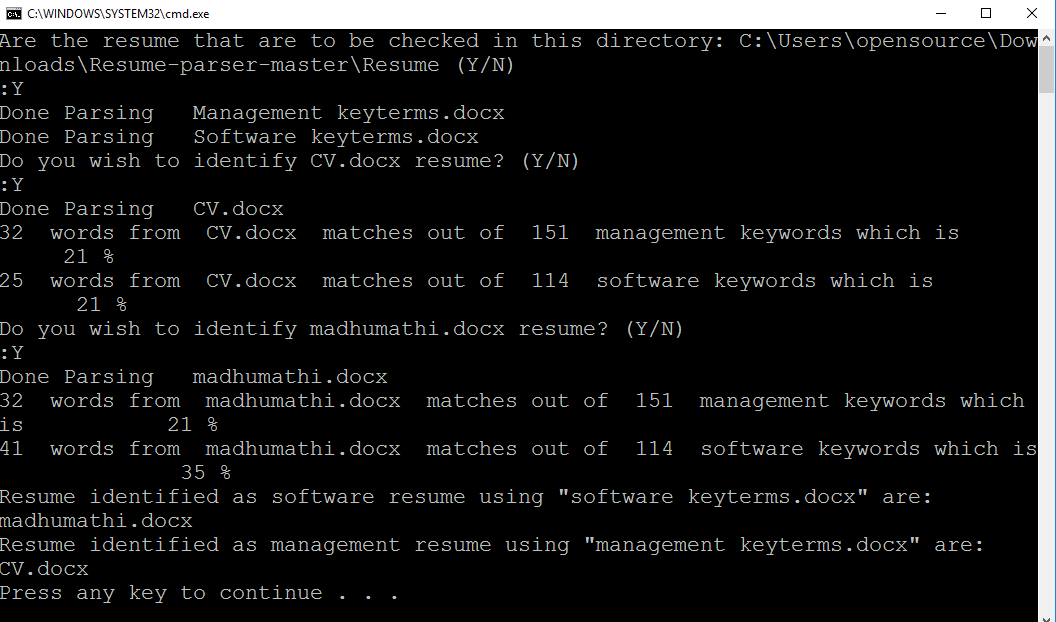
I took Management and Software example resumes, so we need to Doc file for saving keywords of particular domain.

* Create “**Management keyterms.docx**” word document for saving keywords of management related resume classification and save it in the same folder.
* Create “**Software keyterms.docx**” word document for saving keywords of software related resume classification and save it in the same folder.
* Write related keywords into the doc file. (I attached sample keyword doc for reference.)

Step 6:

Run Resume\_identifierpython file using “**python** **resume\_identifier.py**”

It will show resumes classification with %.



Step 7: (Optional)

If we have sample resumes for the particular Domain we can use it to generate a keywords using following step.

Run Keyword\_generator python file using “**python** **keywords\_generator.py**”

It will ask for the each resumes classification setup.

