Name: Nai-Fan Chen NetID: nc41
Task1:

On a5KDtrainingV2.txt

('Task1:', [('applicant', 11393), ('and', 2), ('attack', 3107), ('protein', 433), ('car', 4017)])

On a5KDtestingV2.txt

[('applicant', 14389), ('and', 2), ('attack', 4485), ('protein', 462), ('car', 2924)]

Task2:

This is the logistic function with regularization.

$$\mathrm{LLH}(\theta) = \sum_i y^{(i)}(x^{(i)}\theta) - \log\left(1 + \exp\left\{x^{(i)}\theta\right\}\right) - \lambda\,||\theta||_2$$

This is the derivative function.

$$\frac{\partial f}{\partial r_{j'}} = -x_{i,j'}y_i + x_{i,j'}\left(\frac{e^{\theta}}{1+e^{\theta}}\right) + 2\gamma r_{j'}$$

Update:

Weight -= learning_rate*gradient.

Keep updating until the difference of the loss between every two iterations becomes very small.

On a5KDtrainingV2.txt:

obesity
obese
morbidly
overweight
movi
bariatric
orlistat
hyperphagia
ewl
snacking
sadi
bmi
plication
malabsorptive
worksite
weight
tssc
kj
almonds
tiffin
wize
childcare
kilograms
sibutramine
teenagers
agb
kindergarten
philips
vbg
jib
seminars
overeating
roux
polycystic
hypocaloric
inflating
abstract
satiation
ovary
hypothalamic

tripled
kcal
pizza
mentors
soda
outdoors
banding
iom
idf
jejunal

On a5KDtestingV2:

obesity
obese
bmi
morbid
weight
morbidly
eating
ghrelin
overweight
lifestyle
bariatric
dietitian
epidemic
banding
dispersion
orlistat
norwood
sleeve
flegal
buffet
calmm
leicester
transection
labs
glycaemia
alarming
advice
binge
ameliorated
anovulation
intragastric

hypothesizes
graders
fructose
leptin
tv
lean
metabolically
jejunum
endocannabinoids
employers
anecdotally
television
wanting
enteroendocrine
depots
pyy
putative
clapp
learners

Task3

The weight is trained on a5KDtrainingV2.txt
Predict on a5KDtestingV2.txt

F1:0.600522193211

The F1 score is lower than the small dataset because there are many '0' labels in the big training data. Therefore, the obesity data is not large enough to predict whether it is obesity or not. On the other hand, the not obesity data is very large and we can train our model to predict the sample that is not obesity very accurately. That is to say, the data is very unbalanced. If our data can contain more obesity samples, the model can predict it better. Moreover, because of the skewed data, the machine just predicts that all data is not obesity and the accuracy will be pretty well. That is, the loss will be still small if we have false negative since the positive obesity sample in the training data is the very small partition.

False Positive example:
'0NCT02347267'
'0NCT01150981'
'0NCT00564798'

The above three cases contain the words like 'obesity', 'calories'... The weight of each docum ent will be high and fool the machine to get the positive result. These documents just describ e something else but contain the word with heavy weight. Hence, the model will get confused whether it is obesity or not.

The weight is trained on a5KDtestingV2.txt
Predict on a5KDsmallsetV2.txt

F1:0.8688569101276133

The F1 score is higher than the big training set since the data is not skewed seriously but a little. Due to the reason, the true positive is much larger than the false positive number. That is, the model is accurate in small dataset.

False Positive example:
'0NCT02574949'
'0NCT02318745'
'0NCT02423304'

The reason is the same I write above. Although I might be able to get rid of these case by picking an appropriate threshold, it is difficult to cover them all because TFIDF is the only weight we can use to predict our result. TFIDF just use the frequency of words to compute the score but do not care about context. If we can use RNN or LSTM, we might be able to get better result.