Κ23α - Ανάπτυξη Λογισμικού Για Πληροφοριακά Συστήματα

Χειμερινό Εξάμηνο 2020 – 2021

Καθηγητής Ι. Ιωαννίδης

Άσκηση 1 – Παράδοση: Δευτέρα 21 Δεκεμβρίου 2020

Υλοποίηση αρνητικής συσχέτισης προϊόντων

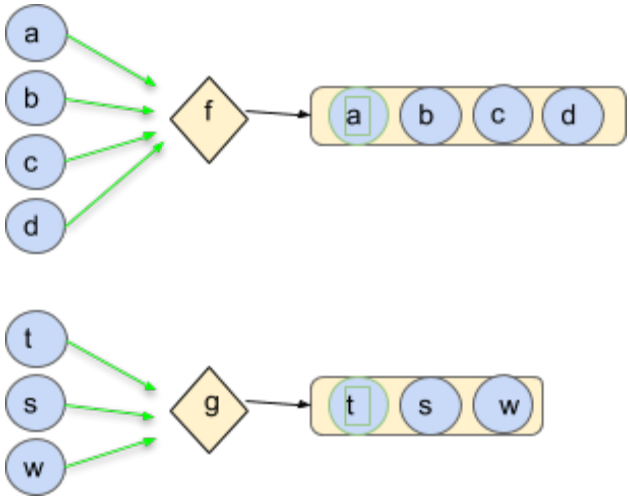
Έχοντας υλοποιήσει τις κλίκες για τα προϊόντα που ταιριάζουν μεταξύ τους, η υλοποίηση θα προχωρήσει στην επεξεργασία προϊόντων που δεν ταιριάζουν μεταξύ τους. Συνεχίζουμε να θεωρούμε ότι κάθε spec θα αντιστοιχιστεί με ένα id (έστω a, b, c,..). Κάθε id θα δείχνει κάθε στιγμή σε ένα σύνολο από άλλα id με τα οποία ταιριάζει, αλλά και πιθανώς σε άλλα σύνολα με τα οποία δεν ταιριάζει.

Η επεξεργασία γραμμών αρνητικής συσχέτισης, δηλαδή του τύπου (a, b, 0) θα πρέπει να λαμβάνει υπόψη της ότι δεν αρκεί να καταγραφεί η αρνητική συσχέτιση a, b, αλλά και το ότι θα πρέπει επαγωγικά να καταγραφεί και αρνητική συσχέτιση μεταξύ της κλίκας στην οποία ανήκει το a και της κλίκας στην οποία ανήκει το b.

Συγκεκριμένα θα ισχύσει ο παρακάτω πίνακας

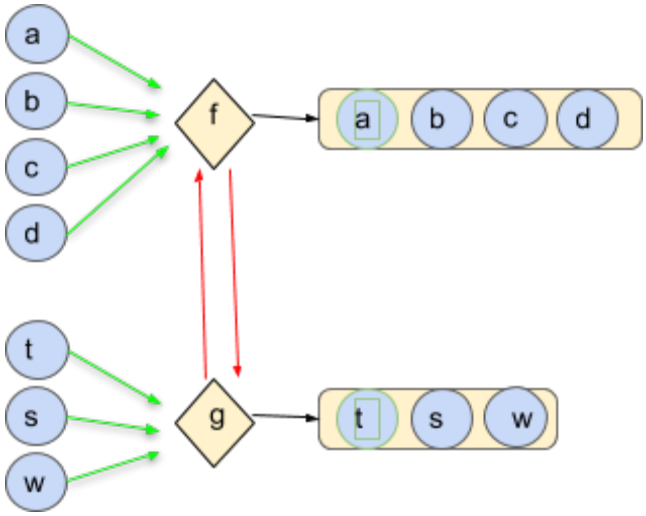
Υπάρχουσα κατάσταση	a, b, 1	a, b, 1	a, b, 0	a, b, 0
Νέα γραμμή	a, c, 1	a, c, 0	a, c, 0	a, c, 1
Επεξεργασία	b, c, 1	b, c, 0	?	b, c, 0

Ο πίνακας αυτός μπορεί να εφαρμοστεί και στη γενικότητα των κλικών. Έστω η εξής



κατάσταση:

Αν σε αυτή την κατάσταση, αναγνωστεί γραμμή π.χ (c, s, 0), τότε θα πρέπει να δημιουργηθεί αρνητική συσχέτιση μεταξύ όλων των μελών της μιας κλίκας με όλα τα μέλη της άλλης. Ένας απλός τρόπος να συμβολιστεί αυτό είναι μέσω της απευθείας αρνητικής συσχέτισης μεταξύ των κλικών, δηλαδή:



Όπως φαίνεται και στο σχήμα, θα πρέπει να υπάρχει αρνητική συσχέτιση τόσο από το σύνολο f προς το σύνολο g, όσο και για την αντίστροφη κατεύθυνση.

Η υλοποίησή σας, θα πρέπει να μπορεί να διαχειριστεί περισσότερες της μίας αρνητικές συσχετίσεις που να εμπλέκουν οποιαδήποτε κλίκα.

Μηχανική Μάθηση

Ένας δυαδικός ταξινομητής είναι μία συνάρτηση που ταξινομεί στιγμιότυπα σε δύο κλάσεις (π.χ. 0 και 1). Η μηχανική μάθηση μας δίνει μεθόδους εκμάθησης ενός τέτοιου ταξινομητή με τρόπο αυτόματο, από ένα σύνολο παραδειγμάτων της μορφής (x,y), δηλαδή ένα σύνολο από ζευγάρια εισόδου-εξόδου. Είναι σύνηθες ο ταξινομητής αυτός να έχει κάποια παραμετρική μορφή φ(.,w) στην οποία περίπτωση το πρόβλημα εκμάθησης γίνεται ένα πρόβλημα βελτιστοποίησης ως προς τις παραμέτρους w κάποιας συνάρτησης σφάλματος L. Η συνάρτηση L που χρησιμοποιείται για δυαδική ταξινόμηση είναι συνήθως η cross-entropy (ή binary cross-entropy). Ένας καλός ταξινομητής πρέπει να μπορεί να ταξινομήσει σωστά νέα δεδομένα που δεν έχει επεξεργαστεί κατά τη διάρκεια εκμάθησης. Αυτό σημαίνει πως πολλές φορές ένας ταξινομητής που μαθαίνει να ταξινομεί καλά το σύνολο δεδομένων εκμάθησης μπορεί να μην είναι καλός ταξινομητής (καθώς μπορεί να μην γενικεύει καλά). Για αυτό το λόγο συνηθίζεται να κρατάμε ένα μικρό μέρος του συνόλου εκμάθησης ξεχωριστά, το οποίο δεν χρησιμοποιείται για την εκπαίδευση του ταξινομητή, άλλα χρησιμοποιείται για την αξιολόγηση ταξινομητών που εκπαιδεύουμε στα υπόλοιπα δεδομένα, με σκοπό να επιλέξουμε τον ταξινομητή που γενικεύει καλύτερα.

Ζητούμενο στην άσκηση είναι να βρεθεί για κάθε πιθανό ζεύγος από προϊόντα το εάν σχετίζονται (1) ή όχι (0). Για ένα μικρό υποσύνολο των ζευγών αυτή ή πληροφορία δίνεται στο αρχείο W. Θα πρέπει να χρησιμοποιήσετε τα δεδομένα αυτά για να φτιάξετε έναν ταξινομητή με τον οποίο θα πάρετε την ζητούμενη πληροφορία για όλα τα υπόλοιπα ζεύγη. Φυσικά, θα πρέπει να χρησιμοποιήσετε όλη τη διαθέσιμη πληροφορία που δίνεται για κάθε προϊόν και είναι επιτρεπτό να κάνετε κάποια προεπεξεργασία στα δεδομένα έτσι ώστε να τα φέρετε σε μία κατάλληλη αναπαράσταση για είσοδο στον ταξινομητή σας.

Προεπεξεργασία - Bag of Words

Το μοντέλο Bag-Of-Words (BOW) είναι μία αναπαράσταση που μετατρέπει κείμενο σε διανύσματα σταθερού μήκους μετρώντας πόσες φορές εμφανίζεται κάθε λέξη στο κείμενο. Η διαδικασία αυτή αναφέρεται και ως διανυσματοποίηση (vectorization).

Παράδειγμα:
Έστω τα εξής 3 κείμενα

- the quick fox jumped
- the quick cat sat
- the dog jumped at the fox

Βήμα 1 - Καθορισμος λεξιλογίου:

Στην αρχή καθορίζουμε το λεξιλόγιο, το οποίο είναι το σύνολο όλων των λέξεων που βρίσκονται στο σύνολο των κειμένων. Οι μόνες λέξεις που βρίσκονται στα 3 κείμενα είναι οι εξής:
the, quick, fox, jumped, cat, sat, dog, at

Βήμα 2 - Μέτρηση
Προκειμένου να ολοκληρώσουμε τη διανυσματοποίηση, θα πρέπει να μετρηθεί η παρουσία των λέξεων στα κείμενα:

Κείμενο	the	quick	fox	jumped	cat	sat	dog	at
1	1	1	1	1	0	0	0	0
2	1	1	0	0	1	1	0	0
3	2	0	1	1	0	0	1	1

Τώρα τα 3 κείμενα μπορούν να μετατραπούν σε διανύσματα διάστασης 8:

- the quick fox jumped [1, 1, 1, 1, 0, 0, 0, 0]
- the quick cat sat [1, 0, 0, 0, 1, 1, 0, 0]
- the dog jumped at the fox [2, 0, 1, 1, 0, 0, 1, 1]

Η τεχνική αυτή αγνοεί τη γραμματική, αλλά και τη σειρά των λέξεων, αλλά προσφέρει μία πιο συμπυκνωμένη μορφή εισόδου για χρήση.

Βελτιώσεις στο BOW

Προκειμένου να μειωθούν οι διαστάσεις των διανυσμάτων που παράγονται, θα πρέπει να μην υπάρχουν διπλότυπες λέξεις με πεζά και κεφαλαία. Συνεπώς αρχικά θα πρέπει όλες οι λέξεις να μετατραπούν σε πεζές (κατάργηση κεφαλαίων). Επίσης τα σημεία στίξης προσθέτουν επιπλέον διαστάσεις και θα πρέπει να αγνοηθούν.
Μία καλή τεχνική είναι να αγνοηθούν κοινές λέξεις όπως άρθρα (the, a, ...), αντωνυμίες (this, who, ...), προθέσεις (in, at, …). Οι λέξεις αυτές ονομάζονται stopwords. Για παράδειγμα στο λεξιλόγιο του παραδείγματος, οι λέξεις 'the' και 'at' είναι stopwords και μπορούν να αγνοηθούν αφού δεν προσφέρουν ιδιαίτερη διαφοροποιηση μεταξύ των κειμένων. Έτσι τα κείμενα θα έχουν τις εξής μετρήσεις:

Κείμενο	quick	fox	jumped	cat	sat	dog
1	1	1	1	0	0	0
2	1	0	0	1	1	0
3	0	1	1	0	0	1

και θα αντιστοιχούν στα εξής διανύσματα διάστασης 6 πλέον:

- the quick fox jumped [1, 1, 1, 0, 0, 0]
- the quick cat sat [0, 0, 0, 1, 1, 0]
- the dog jumped at the fox [0, 1, 1, 0, 0, 1]

Μπορείτε να χρησιμοποιήσετε stopwords από το [1].

TF-IDF

Η τεχνική TF-IDF (Text Frequency - Inverse Document Frequency) είναι μία μετρική που έχει στόχο να δείξει πόσο σημαντική είναι μία λέξη σε ένα κείμενο το οποίο ανήκει σε μία ευρύτερη συλλογή. Η τιμή tf-idf αυξάνει ανάλογα με τον αριθμό των εμφανίσεων ενός όρου στο κείμενο και μειώνεται με τον αριθμό των κειμένων στη συλλογή που περιέχουν τον όρο. Ο υπολογισμός αυτός βοηθάει να υπολογιστεί και η σημαντικότητα εμφάνισης σπάνιων λέξεων σε ένα κείμενο (για παράδειγμα η λέξη 'the' μπορεί να εμφανίζεται σχεδόν σε κάθε κείμενο αρκετές φορές, αλλά δεν προσφέρει ιδιαίτερη πληροφορία).
Ο όρος tf θα είναι το κλάσμα εμφάνισης κάθε λέξης στο κείμενο προς τον συνολικό αριθμό λέξεων του κειμένου.
Συνεχίζοντας το προηγούμενο παράδειγμα, θα έχουμε (μετά την αφαίρεση των stopwords):

Κείμενο	quick	fox	jumped	cat	sat	dog
1	⅓	⅓	⅓	0	0	0
2	⅓	0	0	⅓	⅓	0
3	0	⅓	⅓	0	0	⅓

Ο όρος idf υπολογίζεται μία φορά για κάθε λέξη για όλη τη συλλογή.
$\mathrm{idf}_t = \log \frac{n}{n_t}$, όπου n συνολικός αριθμός κειμένων της συλλογής και n_t ο αριθμός κειμένων της συλλογής που περιέχουν τον συγκεκριμένο όρο.
Για παράδειγμα, ο όρος 'quick' θα έχει τιμή
$idf_t = \log \frac{3}{2} = 0.18$
και ο όρος 'cat' αντίστοιχα:
$idf_t = \log \frac{3}{1} = 0.48$
Ο τελικός πίνακας μετά την εφαρμογή tf-idf θα είναι:

Κείμενο	quick	fox	jumped	cat	sat	dog
idf	0.18	0.18	0.18	0.48	0.48	0.48

1	⅓*0.18	⅓*0.18	⅓*0.18	0*0.48	0*0.48	0*0.48
2	⅓*0.18	0*0.18	0*0.18	⅓*0.48	⅓*0.48	0*0.48
3	0*0.18	⅓*0.18	⅓*0.18	0*0.48	0*0.48	⅓*0.48

ή μετά τις πράξεις:

Κείμενο	quick	fox	jumped	cat	sat	dog
1	0.0587	0.0587	0.0587	0	0	0
2	0.0587	0	0	0.16	0.16	0
3	0	0.0587	0.0587	0	0	0.16

Έχουμε δηλαδή μία αποτελεσματικότερη απεικόνιση της σημασίας των εμφανιζόμενων λέξεων, ανάλογα με το πόσο συχνά ή σπάνια εμφανίζονται στα κείμενα. Σε περίπτωση που θέλουμε να μειώσουμε τις διαστάσεις (λέξεις) που θα χρησιμοποιήσουμε στα μοντέλα εκτίμησης που θα χρησιμοποιήσουμε, μπορούμε να αποβάλουμε τις λέξεις με το μικρότερο idf (συχνές εμφανίσεις, μικρή συνεισφορά πληροφορίας).

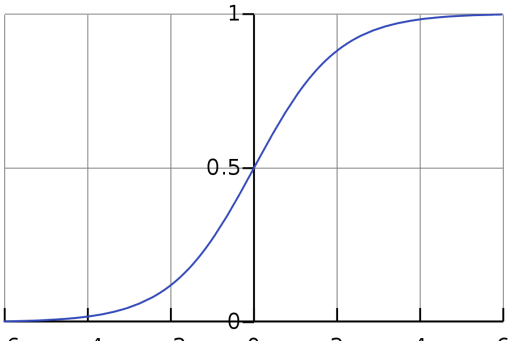
Λογιστική παλινδρόμηση (Logistic Regression)

Η λογιστική παλινδρόμηση είναι ένας αλγόριθμος ταξινόμησης μηχανικής μάθησης που χρησιμοποιείται για να εκτιμήσει σε ποια από τις ορισμένες τάξεις ανήκει μία παρατήρηση/δείγμα. Παραδείγματα χρήσης είναι να ο διαχωρισμός email σε spam/ham, διάκριση καλοήθειας/κακοήθειας σε ακτινογραφίες, κ.ο.κ. Η λογιστική παλινδρόμηση που θα χρησιμοποιήσουμε θα είναι δυαδική.

Το μοντέλο λογιστικής παλινδρόμηση είναι παρόμοιο με το μοντέλο γραμμικής παλινδρόμησης, αλλά με μία πολυπλοκότερη συνάρτηση σφάλματος. Η συνάρτηση σφάλματος είναι η λογιστική συνάρτηση (logistic or sigmoid function) αντί για μία γραμμική συνάρτηση.

Προκειμένου να αντιστοιχίσουμε τις τιμές πρόβλεψης σε πιθανότητες, χρησιμοποιούμε τη λογιστική συνάρτηση. Η συνάρτηση αντιστοιχίζει κάθε τιμή πραγματικού αριθμού σε μία άλλη τιμή μεταξύ 0 και 1.

$\sigma : \mathbb{R} \to (0,1)$ όπου $\sigma(t) = \frac{1}{1+e^{-t}}$



https://commons.wikimedia.org/wiki/File:Logistic-curve.svg#/media/File:Logistic-curve.svg

Θα υλοποιήσουμε τη λογιστική παλινδρόμηση μιας εξαρτημένης μεταβλητής y στο σύνολο των εξαρτημένων μεταβλητών $\mathbf{x} = (x_1, x_2, \ldots, x_r)$, όπου r είναι το πλήθος των διαστάσεων του διανύσματος εισόδου. Ξεκινάμε με τις γνωστές τιμές των διανυσμάτων εισόδου $\mathbf{x_i}$ και την αντίστοιχη πραγματική απόκριση y_i, για κάθε γνωστή παρατήρηση $i = 1, 2, \ldots, n$.

Στόχος είναι να βρεθεί η συνάρτηση λογιστικής παλινδρόμησης $p(\mathbf{x})$, ώστε οι αποκρίσεις της πρόγνωσης $p(\mathbf{x_i})$ να είναι όσο εγγύτερα γίνεται στην πραγματική απόκριση y_i για κάθε παρατήρηση $i = 1, 2, \ldots, n$. Υπενθυμίζεται ότι η πραγματική απόκριση μπορεί να είναι μόνο 0 ή 1 σε προβλήματα δυαδικής κατηγοριοποίησης. Αυτό σημαίνει ότι κάθε $p(\mathbf{x_i})$ θα πρέπει να είναι κοντά είτε στο 0 είτε στο 1. Αυτός είναι και ο λόγος που χρησιμοποιείται η σιγμοειδής συνάρτηση.

Από τη στιγμή που θα οριστικοποιηθεί η συνάρτηση λογιστικής παλινδρόμησης $p(\mathbf{x})$, μπορούμε να τη χρησιμοποιήσουμε για την πρόγνωση εξόδων για νέες και άγνωστες εισόδους, θεωρώντας ότι οι υποκείμενες μαθηματικές εξαρτήσεις δεν έχουν αλλάξει.

Μεθοδολογία

Η λογιστική παλινδρόμηση είναι ένας γραμμικός ταξινομητής, επομένως θα χρησιμοποιηθεί μία γραμμική συνάρτηση $f(\mathbf{x}) = b + w_1 x_1 + w_2 x_2 \cdots + w_r x_r$ ή $f(\mathbf{x}) = b + \mathbf{w}^{\mathrm{T}} \mathbf{x}$. Οι μεταβλητές $\mathbf{w} = w_1, w_2, \ldots, w_r$ είναι οι εκτιμητές των συντελεστών παλινδρόμησης (βάρη πρόγνωσης ή απλώς συντελεστές).

Η συνάρτηση λογιστικής παλινδρόμησης $p(\mathbf{x})$ είναι η σιγμοειδής συνάρτηση του $f(\mathbf{x})$: $p(\mathbf{x}) = \frac{1}{1+e^{-f(x)}}$ ή $p(\mathbf{x}) = \sigma(f(\mathbf{x}))$

Επομένως η τιμή της κυμαίνεται μεταξύ 0 και 1 και ερμηνεύεται ως η πρόγνωση πιθανότητας ότι η έξοδος για ένα δεδομένο $\mathbf{x}$ να ισούται με 1. Αντιστρόφως, το $1 - p(\mathbf{x})$ είναι η πιθανότητα ότι η έξοδος ισούται με 0.

Η λογιστική παλινδρόμηση καθορίζει τα βέλτιστα βάρη πρόγνωσης $b, w_1, w_2, \ldots, w_r$, ώστε η συνάρτηση $p(\mathbf{x})$ να είναι όσο πιο κοντά γίνεται στις πραγματικές αποκρίσεις $y^{(i)}$, $i = 1, 2, \ldots, n$. Η διαδικασία του υπολογισμού των βέλτιστων βαρών πρόγνωσης χρησιμοποιώντας τις διαθέσιμες παρατηρήσεις ονομάζεται εκπαίδευση του μοντέλου (training, fitting).

Για την εύρεση των βέλτιστων βαρών, μεγιστοποιείται η συνάρτηση LLF (log-likelihood function) για όλες τις παρατηρήσεις $i = 1, 2, \ldots, n$. Η μέθοδος αυτή είναι η εκτίμηση μέγιστης πιθανότητας.

Συνάρτηση σφάλματος

Η συνάρτηση σφάλματος στη λογιστική παλινδρόμηση ορίζεται ως εξής:

$$L = \begin{cases} -\log\left(p(\mathbf{x})\right) & , y = 1 \\ -\log\left(1 - p(\mathbf{x})\right) & , y = 0 \end{cases}$$

ή εναλλακτικά $L = -y \log\left(p(\mathbf{x})\right) - (1-y) \log\left(1 - p(\mathbf{x})\right)$

Επειδή $p(\mathbf{x}) = \sigma(f(\mathbf{x}))$

$$L(\mathbf{w}, \mathrm{b}) = -y \log\left(\sigma\left(\mathbf{w}^{\mathrm{T}}\mathbf{x}+\mathrm{b}\right)\right) - (1-y) \log\left(1 - \sigma\left(\mathbf{w}^{\mathrm{T}}\mathbf{x}+\mathrm{b}\right)\right)$$

Η συνάρτηση αυτή ονομάζεται και log loss ή cross entropy.

$$J(\mathbf{w}, b) = -\frac{1}{n} \sum_{i=1}^{n} \left[y^{(i)} \log\left(\sigma\left(\mathbf{w}^{\mathrm{T}}\mathbf{x}+\mathrm{b}\right)\right) + \left(1 - y^{(i)}\right) \log\left(1 - \sigma\left(\mathbf{w}^{\mathrm{T}}\mathbf{x}+\mathrm{b}\right)\right) \right]$$

Για την ελαχιστοποίηση του σφάλματος, δηλαδή της τιμής της LLF, ακολουθείται ο αλγόριθμος της απότομης καθόδου ή αλλιώς ο αλγόριθμος ελάττωσης της παραγώγου (gradient descent). Στην περίπτωση αυτή η παράγωγος της λογιστικής συνάρτησης βρίσκεται εύκολα ως εξής:

$$\sigma'(z) = \sigma(z)(1 - \sigma(z))$$

που οδηγεί στην εξής παραγώγιση της συνάρτησης σφάλματος

$$\frac{\partial}{\partial w_j} J(\mathbf{w}, b) = \sum_{i=1}^{n} \left(\sigma\left(\mathbf{w}^{\mathrm{T}}\mathbf{x}^{(\mathrm{i})} + b\right) - y^{(i)}\right) x_j^{(i)}$$

όπου

- $\frac{\partial}{\partial w_j} J(\mathbf{w}, b)$ είναι η παράγωγος του σφάλματος ως προς το βάρος w_j
- $y^{(i)}$ είναι η πραγματική τιμή της κλάσης (0 ή 1) για την παρατήρηση i
- $\sigma\left(\mathbf{w}^{\mathrm{T}}\mathbf{x}^{(\mathrm{i})} + b\right)$ είναι η πρόγνωση του μοντέλου για την παρατήρηση i
- $x_j^{(i)}$ η είσοδος για την παρατήρηση i, ως προς τη διάσταση j που αντιστοιχεί στο βάρος w_j

Το μήκος της διαφοροποίησης για κάθε w_j εξαρτάται από την κλίση της καμπύλης $\frac{\partial}{\partial b_j} J(\mathbf{w}, b)$ που πολλαπλασιάζεται με τον ρυθμό εκμάθησης (learning rate) η. Επομένως η αλλαγή που θα γίνεται για κάθε βάρος θα είναι $\mathbf{w}^{t+1} = \mathbf{w}^{t} - \eta \nabla J(\mathbf{w}, b)$

Η διαδικασία αυτή μπορεί να επαναλαμβάνεται για πολλά βήματα. Ένας τρόπος διακοπής της διαδικασίας αυτής είναι όταν $\mathbf{w}^{t+1} - \mathbf{w}^{t} < \varepsilon$, όπου ε ένας πολύ μικρός αριθμός.

Όταν ολοκληρωθεί η διαδικασία αυτή, το μοντέλο θεωρείται ότι έχει εκπαιδευτεί, και μπορούμε να προχωρήσουμε στο βήμα ελέγχου (testing).

Στο βήμα ελέγχου, ανατίθενται ετικέτες κλάσης (0 ή 1) σε αντικείμενα, για τα οποία δεν είναι δεδομένη η κλάση τους, σύμφωνα με την πιθανότητα που υπολογίζει το εκπαιδευμένο μοντέλο λογιστικής παλινδρόμησης.

Παράδοση εργασίας

Προθεσμία παράδοσης: 21/12/2020
Γλώσσα υλοποίησης: C / C++ χωρίς χρήση stl.
Περιβάλλον υλοποίησης: Linux (gcc > 5.4+).

Παραδοτέα: Η παράδοση της εργασίας θα γίνει με βάση το τελευταιο commit πριν την προθεσμία υποβολής στο git repository σας. **Η χρήση git είναι υποχρεωτική**.
Επιπλέον, εκτός από τον πηγαίο κώδικα, θα παραδώσετε μια σύντομη αναφορά, με τις σχεδιαστικές σας επιλογές καθώς και να εφαρμόσετε ελέγχους ως προς την ορθότητα του λογισμικού με τη χρήση ανάλογων βιβλιοθηκών (Software testing).

Αναφορές

[1] Common English Words as csv: https://www.textfixer.com/tutorials/common-english-words.txt