



Πανεπιστήμιο Πελοποννήσου
Τεχνητή Νοημοσύνη
Εαρινό Εξάμηνο 2024

Αναφορά Δεύτερης Εργασίας

Γιαννόπουλος Γεώργιος
ΑΜ:2022202000039

Γιαννόπουλος Ιωάννης
ΑΜ: 2022201900032

Ιούνιος, 2024

Περιεχόμενα

1	Εισαγωγή	3
2	Λεπτομέρειες Υλοποίησης	3
2.1	Δομή Φακέλου	3
2.2	Bayes Networks Module	4
2.3	Probability Distribution Module	4
2.4	Data Module	4
2.4.1	Κλάση NaiveBayesClassifier	5
2.5	Main	5
2.6	Εκτέλεση Προγράμματος	6
3	Δίκτυο Bayes	6
3.1	Κόμβοι και Αχμές Δικτύου Bayes	6
3.2	Υπολογισμός Πιθανοτήτων	8
4	Φίλτρο Ανίχνευσης Spam Μηνυμάτων με Χρήση του Αλ- γορίθμου Naive Bayes	9
4.1	Βήματα Υλοποίησης Ταξινόμητη	9
4.2	Αξιολόγηση Αρχικού Ταξινόμητη	10
4.3	Αξιολόγηση Ταξινόμητη με Εξομάλυνση Laplace	10
4.4	Αξιολόγηση Ταξινόμητη με Αποφυγή Σφάλματος Ανεπάρκειας	10
4.5	Απόδειξη	11
5	Επίλογος	12

1 Εισαγωγή

Στα πλαίσια της δεύτερης εργασίας του μαθήματος *Τεχνητής Νοημοσύνης*, θα ασχοληθούμε με την κατασκευή και την εφαρμογή δικτύων *Bayes*, καθώς και με τη χρήση του αλγορίθμου *Naive Bayes* ως ταξινομητή για την επίλυση πραγματικών προβλημάτων.

Οι τεχνικές *Bayesian networks* και *Naive Bayes* αποτελούν θεμελιώδεις μεθόδους στον τομέα της *Τεχνητής Νοημοσύνης* και της *Μηχανικής Μάθησης*. Μέσω αυτών, θα αποκτήσουμε βαθύτερη κατανόηση των πιθανολογικών μοντέλων και θα αναπτύξουμε πρακτικές δεξιότητες που είναι απαραίτητες για την αντιμετώπιση σύνθετων προβλημάτων.

Με αυτήν την εργασία, στοχεύουμε όχι μόνο στην απόκτηση θεωρητικής γνώσης αλλά και στην ανάπτυξη πρακτικών δεξιοτήτων που θα μας επιτρέψουν να εφαρμόσουμε τις τεχνικές αυτές σε πραγματικά προβλήματα, προσφέροντας λύσεις με σημαντική προστιθέμενη αξία.

2 Λεπτομέρειες Υλοποίησης

Αυτή η ενότητα παρέχει μια επισκόπηση της οργάνωσης του κώδικα και των αρχείων που χρησιμοποιούνται για την υλοποίηση της εργασίας.

2.1 Δομή Φακέλου

Η δομή που ακολουθείτε είναι η εξής:

```
AI_Project2
├── bayes_networks
│   ├── bayes_net.py
│   ├── bayes_network_utils.py
│   └── bayes_node.py
├── data
│   ├── clean_data.py
│   ├── download_data.py
│   ├── naive_bayes_classifier.py
│   └── split_data.py
├── probability_distribution
│   └── probdist.py
└── main.py
```

2.2 Bayes Networks Module

Στον φάκελο¹ αυτόν περιέχονται τα εξής αρχεία:

- `bayes_net.py`
 - Αυτή η ενότητα ορίζει την κλάση **BayesNet** για την αναπαράσταση ενός δικτύου Bayes.
- `bayes_network_utils.py`
 - Αυτή η ενότητα περιέχει συναρτήσεις για την πραγματοποίηση ερωτημάτων σε δίκτυα Bayes.
- `bayes_node.py`
 - Αυτή η ενότητα ορίζει την κλάση **BayesNode** για την αναπαράσταση κόμβων σε ένα δίκτυο Bayes.

2.3 Probability Distribution Module

Αυτή η ενότητα ορίζει την κλάση **ProbDist** για την αναπαράσταση μιας διακριτής κατανομής πιθανοτήτων και μια συνάρτηση χρησιμότητας για το χειρισμό συμβάντων.

2.4 Data Module

Ο φάκελος αυτός περιέχει κώδικα που μας δόθηκε και πιο συγκεκριμένα τα παρακάτω αρχεία:

- `download_data.py`
 - Ενότητα για λήψη και εξαγωγή του συνόλου δεδομένων ηλεκτρονικού ταχυδρομείου.
- `clean_data.py`
 - Αυτή η ενότητα περιέχει συναρτήσεις για τον καθαρισμό των δεδομένων.
- `split_data.py`

¹Τα αρχεία είχαν δοθεί μαζί με την εκφώνηση της εργασίας και έχουμε προστέσει υποδείξεις τύπου στις συναρτήσεις με το `typing module` της Python για ευκολία στην χρήση τους

- Ενότητα για το διαχωρισμό των μηνυμάτων ηλεκτρονικού ταχυδρομείου και των ετικετών σε σύνολα εκπαίδευσης και ελέγχου.

Τέλος περιέχει την κλάση **NaiveBayesClassifier**, που δημιουργήσαμε για να υλοποιήσουμε τον αλγόριθμο **Naive Bayes**.

2.4.1 Κλάση NaiveBayesClassifier

1. Αρχικοποίηση του ταξινομητή Naive Bayes.
 - Αρχικοποιεί τα χαρακτηριστικά που χρειάζονται για τον υπολογισμό των πιθανοτήτων.
2. Εκπαίδευση του ταξινομητή.
 - Η μέθοδος **train** της κλάσης, χρησιμοποιείται για την εκπαίδευση του μοντέλου μας. Δέχεται ως παραμέτρους μία λίστα από email εκπαίδευσης (`emails_train`), μία λίστα των κατηγοριών τους (`y_train`) και την λογική παράμετρο `laplace_smoothing`.
3. Πρόβλεψη ετικετών .
 - Η μέθοδος **predict** της κλάσης, χρησιμοποιείται για την πρόβλεψη νέων ετικετών από το σύνολο ελέγχου των ηλεκτρονικών μηνυμάτων. Ως παραμέτρους δέχεται μία λίστα από emails (`emails_test`), και την λογική παράμετρο `prevent_underflow`. Επιστρέφει μία λίστα από αλφαριθμητικά που περιέχουν την κατηγορία που πρόβλεψε, spam ή ham.
4. Αξιολόγηση μοντέλου
 - Χρήση της μετρικής αξιολόγησης **ακρίβειας** με την μέθοδο **accuracy**. Ως ορίσματα λαμβάνει τις ετικέτες από το σετ ελέγχου (πραγματικές κατηγορίες) και τις κατηγορίες που πρόβλεψε με την μέθοδο **predict**. Επιστρέφει την ακρίβεια που είναι μία πραγματική τιμή στο πεδίο $[0, 1]$

2.5 Main

Το κύριο πρόγραμμα στο οποίο πραγματοποιούνται όλα τα ερωτήματα της εργασίας, καθώς και οι αρχικοποιήσεις των κλάσεων και η χρήση των συναρτήσεων και των μεθόδων.

2.6 Εκτέλεση Προγράμματος

Το πρόγραμμά μας για να τρέξει απαιτεί να έχουμε μεταβεί στην τοποθεσία που είναι αποθηκευμένος ο φάκελος **AI_Project2**

```
cd AI_Project2
python main.py
```

Listing 1: Εντολές Εκτέλεσης Προγράμματος μέσα από τερματικό

Η μόνη βιβλιοθήκη που δεν είναι προεγκατεστημένη είναι η *NumPy* η οποία μπορεί να εγκατασταθεί εύκολα με την παρακάτω εντολή:

```
pip install numpy
```

Listing 2: Εγκατάσταση βιβλιοθήκης NumPy

3 Δίκτυο Bayes

Δεδομένων των πληροφοριών που μας δίνονται από την εκφώνηση της εργασίας, για την δημιουργία του δικτύου Bayes θα ορίσουμε τους κόμβους και τις σχέσεις τους.

3.1 Κόμβοι και Ακμές Δικτύου Bayes

Οι εξαρτήσεις μεταξύ αυτών των μεταβλητών αποτυπώνονται με τις ακμές του δικτύου Bayes. Οι εξαρτήσεις όπως περιγράφονται μπορούν να αναπαρασταθούν ως εξής:

- Η μεταβλητή *Taksidevei* επηρεάζει την πιθανότητα μιας απάτης (*Apati*). Συγκεκριμένα, η πιθανότητα απάτης είναι διαφορετική όταν ο κάτοχος της κάρτας ταξιδεύει.
- Η μεταβλητή *Apati* επηρεάζει την πιθανότητα μιας αγοράς εξωτερικού (*AgoraEksoterikou*).
- Η μεταβλητή *DiatheteiIpologisti* επηρεάζει την πιθανότητα μιας αγοράς μέσω Διαδικτύου (*AgoraDiadiktou*).
- Η μεταβλητή *AgoraDiadiktou* επηρεάζει την πιθανότητα απάτης (*Apati*).

- Η μεταβλητή *DiatheteiIpologisti* επηρεάζει την πιθανότητα αγοράς σχετικής με υπολογιστή την περασμένη εβδομάδα (*AgoraSxetikiMeIpologisti*).

Βάσει αυτών των πληροφοριών, η δομή του δικτύου Bayes μπορεί να διαμορφωθεί ως εξής:

1. Taksidevei → Apati

- Όταν ο κάτοχος της κάρτας ταξιδεύει, οι πιθανότητες απάτης είναι υψηλότερες (1%) από ό,τι όταν δεν ταξιδεύει (0.4%). Επομένως, το αν ο κάτοχος ταξιδεύει ή όχι **επηρεάζει** άμεσα την πιθανότητα μιας συναλλαγής να είναι απάτη.

2. Apati → AgoraEksoterikou

- Όταν μια συναλλαγή είναι απάτη, η πιθανότητα να είναι αγορά εξωτερικού αυξάνεται, εκτός αν ο κάτοχος ταξιδεύει. Επομένως, η πιθανότητα μιας συναλλαγής να είναι αγορά εξωτερικού **εξαρτάται** από το αν η συναλλαγή είναι απάτη.

3. DiatheteiIpologisti → AgoraDiadiktiou

- Οι πιθανότητες αγοράς μέσω Διαδικτύου **εξαρτώνται** από το αν ο κάτοχος της κάρτας διαθέτει υπολογιστή. Συγκεκριμένα, το 1% των νόμιμων συναλλαγών γίνεται μέσω Διαδικτύου για όσους έχουν υπολογιστή, ενώ το ποσοστό αυτό αυξάνεται στο 2% για συναλλαγές από απάτη. Για όσους δεν διαθέτουν υπολογιστή, το 0.1% των νόμιμων συναλλαγών γίνεται μέσω Διαδικτύου και το 1.1% των συναλλαγών από απάτη.

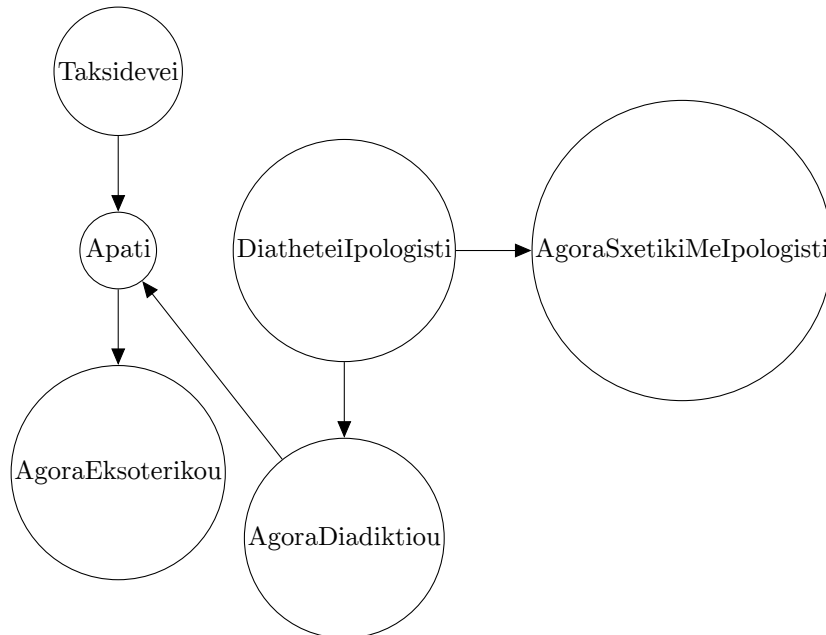
4. AgoraDiadiktiou → Apati

- Οι αγορές που γίνονται μέσω Διαδικτύου είναι πιο πιθανό να είναι απάτες. Έτσι, η πιθανότητα απάτης **επηρεάζεται** από το αν η συναλλαγή γίνεται μέσω Διαδικτύου.

5. DiatheteiIpologisti → AgoraSxetikiMeIpologisti

- Το εάν ο κάτοχος της κάρτας διαθέτει υπολογιστή ή όχι, το προβλέπει εάν την προηγούμενη εβδομάδα υπήρχε αγορά σχετική με υπολογιστή.

Γραφικά το δίκτυο Bayes μπορεί να αναπαρασταθεί ως:



Σχήμα 1: Δίκτυο Bayes για την ανίχνευση απάτης σε συναλλαγές με κάρτα

3.2 Υπολογισμός Πιθανοτήτων

Αφού έχουμε κατασκευάσει το δίκτυο Bayes, μπορούμε να προβούμε στον υπολογισμό πιθανοτήτων για διάφορα σενάρια. Πιο συγκεκριμένα θα επικεντρωθούμε σε δύο σενάρια:

1. $P(\text{Apati} \mid \text{AgoraEksoterikou} = \text{True}, \text{AgoraDiadiktiou} = \text{False}, \text{AgoraSxetikiMeIpologisti} = \text{True})$
2. $P(\text{Apati} \mid \text{Taksidevei} = \text{True}, \text{AgoraEksoterikou} = \text{True}, \text{AgoraDiadiktiou} = \text{False}, \text{AgoraSxetikiMeIpologisti} = \text{True})$

Στην πρώτη περίπτωση υπολογίζουμε την πιθανότητα να έχουμε απάτη δεδομένου ότι:

- είναι αγορά εξωτερικού
- δεν είναι αγορά διαδικτύου
- είναι αγορά σχετική με υπολογιστή

Στην δεύτερη περίπτωση προσθέτουμε και το δεδομένο ότι

- ο κάτοχος ταξιδεύει, βρίσκεται στο εξωτερικό.

Ως αποτέλεσμα λαμβάνουμε:

1. Για την πρώτη περίπτωση:
 - Ψευδές: 0.985, Αληθές: 0.015
2. Για την δεύτερη περίπτωση:
 - Ψευδές 0.99, Αληθές: 0.0099

Παρατηρούμε ότι όταν έχουμε το ενδεχόμενο ο κάτοχος της κάρτας να ταξιδεύει, αυτομάτως να μειώνεται και η πιθανότητα για συναλλαγή απάτης. Η πιθανότητα απάτης μειώνεται ελαφρώς στο **0,99%**, επειδή η μεταβλητή του ταξιδιού επηρεάζει άμεσα την πιθανότητα απάτης καθώς αυτή αυξάνεται όταν είναι αγορά εξωτερικού εκτός και εάν ταξιδεύει. Οπότε αφού ο κάτοχος ταξιδεύει μειώνει την πιθανότητα απάτης. Ενώ στην πρώτη περίπτωση, η πιθανότητα είναι υψηλότερη στο **1,5%**, επειδή η αβεβαιότητα σχετικά με την ταξιδιωτική κατάσταση σε συνδυασμό με τις άλλες συνθήκες υποδηλώνουν υψηλότερο κίνδυνο απάτης.

4 Φίλτρο Ανίχνευσης Spam Μηνυμάτων με Χρήση του Αλγορίθμου Naive Bayes

Στο δεύτερο μέρος της εργασίας υλοποιήσαμε τον αλγόριθμο **Naive Bayes** για την ταξινόμηση μηνυμάτων ηλεκτρονικού ταχυδρομείου σε spam ή μη spam (ham).

4.1 Βήματα Υλοποίησης Ταξινομητή

Για την υλοποίηση του **Naive Bayes Classifier** ακολουθούνται τα εξής βήματα:

1. Κατέβασμα του συνόλου δεδομένων με την χρήση της συνάρτησης `download_dataset`
2. Εξαγωγή των email και των κατηγοριών τους με χρήση της `extract_dataset`
3. Προεπεξεργασία των email με κανονικές εκφράσεις με χρήση της συνάρτησης `clean_emails`
4. Χωρισμός των email και των κατηγοριών τους σε σύνολο εκπαίδευσης και σύνολο ελέγχου με χρήση της `split_data`

5. Αρχικοποίηση του ταξινομητή Naive Bayes χρησιμοποιώντας την κλάση *NaiveBayesClassifier*.
6. Εκπαίδευση του ταξινομητή με τα σύνολα εκπαίδευσης χρησιμοποιώντας την μέθοδο *train*.
7. Πρόβλεψη των κατηγοριών με το σύνολο ελέγχου χρησιμοποιώντας την μέθοδο *predict*.
8. Χρήση της μεθόδου *accuracy* ως μετρική αξιολόγησης του ταξινομητή.

4.2 Αξιολόγηση Αρχικού Ταξινομητή

Στην πιο απλή μορφή του ο ταξινομητής μας καταφέρνει μία ακρίβεια ίση με **0.842²** ή **84.2%**. Αυτό μπορεί να ερμηνευθεί ως ότι στα 100 email έχει καταφέρει να κατατάσσει σωστά **84** και λανθασμένα τα υπόλοιπα **16**. Αν και η ακρίβεια του δεν μπορεί να θεωρηθεί υψηλή αποτελεί μία πάρα πολύ καλή αρχή, καθώς παρά τις απλοποιήσεις που επιφέρει η υπόθεση της ανεξαρτησίας των χαρακτηριστικών, ο Naive Bayes ταξινομητής παραμένει ένας ισχυρός και αποδοτικός αλγόριθμος για την ανίχνευσης spam.

4.3 Αξιολόγηση Ταξινομητή με Εξομάλυνση Laplace

Κατά την διάρκεια της εκπαίδευσης του ταξινομητή ο χρήστης μπορεί να θέσει την παράμετρο *laplace_smoothing* της μεθόδου **train** σε **True** (Αληθές), ώστε να εφαρμοστεί η εξομάλυνση Laplace στο μοντέλο μας. Οπότε αφού θέσουμε την παράμετρο σε αληθές και καλώντας την μέθοδο *accuracy* λαμβάνουμε μία ακρίβεια ίση με **0.931** ή **93.1%**. Φυσικά και αποτελεί σημαντική αύξηση σε σχέση με το προηγούμενο μοντέλο μας, καταφέροντας να μειώσει τις λάθος προβλέψεις από τις 16 στις **7**, 9 προβλέψεις διαφορά.

4.4 Αξιολόγηση Ταξινομητή με Αποφυγή Σφάλματος Ανεπάρκειας

Ακόμα ο χρήστης μπορεί κατά την διάρκεια της πρόβλεψης των κατηγοριών να θέσει την παράμετρο *prevent_underflow* της μεθόδου **predict** σε **True** (Αληθές), ώστε να χρησιμοποιήσει τις λογαριθμικές πιθανότητες για την πρόβλεψη της κλάσης που ανήκει. Οπότε η ακρίβεια του μοντέλου μας, αυξάνεται σε **0.963** ή **96.3%**. Οπότε και κατορθώνουμε να κατατάσσουμε σωστά **96** από τις **100** προβλέψεις.

²Έχει πραγματοποιηθεί στρογγυλοποίηση σε τρία δεκαδικά σημεία

```
Accuracy without Laplace Smoothing and underflow prevention: 0.842
Accuracy with Laplace Smoothing set True and prevent underflow to False: 0.931
Accuracy with Laplace Smoothing set True and prevent underflow to True: 0.963
```

Σχήμα 2: Η ακρίβεια του ταξινομητή με τις διαφορές παραμέτρους

4.5 Απόδειξη

Μας δίνεται ότι ισχύει η ανισότητα αυτή:

$$P(d|C_{\text{ham}}) \times P(C_{\text{ham}}) > P(d|C_{\text{ham}}) \times P(C_{\text{spam}})$$

καθώς και αυτή:

$$P(d|C_{\text{ham}}) \times P(C_{\text{ham}}) < P(d|C_{\text{ham}}) \times P(C_{\text{spam}})$$

Η λογαριθμική συνάρτηση, που χρησιμοποιούμε, είναι γνησίως αύξουσα καθώς έχει βάση > 1 . Αυτό σημαίνει ότι εάν $a > b$, τότε $\log(a) > \log(b)$, και εάν $a < b$, τότε $\log(a) < \log(b)$. Οπότε και οι σχέσεις μας γίνονται:

$$\log(P(d|C_{\text{ham}}) \times P(C_{\text{ham}})) > \log(P(d|C_{\text{ham}}) \times P(C_{\text{spam}}))$$

και

$$\log(P(d|C_{\text{ham}}) \times P(C_{\text{ham}})) < \log(P(d|C_{\text{ham}}) \times P(C_{\text{spam}}))$$

Η σύγκριση μεταξύ των λογαρίθμων παρέμεινε η ίδια με την αρχική. Αυτό σημαίνει ότι εάν ισχύει $P(d|C_{\text{ham}}) \times P(C_{\text{ham}}) > P(d|C_{\text{ham}}) \times P(C_{\text{spam}})$, τότε ισχύει:

$$\log(P(d|C_{\text{ham}}) \times P(C_{\text{ham}})) > \log(P(d|C_{\text{ham}}) \times P(C_{\text{spam}}))$$

Το ίδιο ισχύει και για την άλλη περίπτωση, όπου εάν ισχύει ότι $P(d|C_{\text{ham}}) \times P(C_{\text{ham}}) < P(d|C_{\text{ham}}) \times P(C_{\text{spam}})$, τότε ισχύει:

$$\log(P(d|C_{\text{ham}}) \times P(C_{\text{ham}})) < \log(P(d|C_{\text{ham}}) \times P(C_{\text{spam}}))$$

Οπότε **ισχύει** και επομένως ο αλγόριθμος του Naive Bayes συνεχίζει να είναι σωστός και μετά την εφαρμογή των λογαρίθμων.

5 Επίλογος

Συνοψίζοντας, η εργασία αυτή προσέφερε μια ολοκληρωμένη ανάλυση και εφαρμογή των **δικτύων Bayes** και του αλγορίθμου **Naive Bayes**. Αποτέλεσε σημαντικό βήμα για την υλοποίηση της θεωρίας, που αποκτήθηκε κατά την διάρκεια του μαθήματος, σε μία πρακτική εφαρμογή, σε δύο απλοποιημένα σενάρια.

Καταφέραμε να μοντελοποιήσουμε ένα δίκτυο Bayes για την εξαγωγή συμπερασμάτων σε διάφορα σενάρια, για το αν μία συναλλαγή αποτελεί ενδεχόμενη απάτη. Ακόμα η χρήση του αλγορίθμου Naive Bayes για την κατηγοριοποίηση ηλεκτρονικών μηνυμάτων αποδείχθηκε αποτελεσματική, χάρη στην ικανότητά του να επεξεργάζεται μεγάλες ποσότητες δεδομένων με σχετική απλότητα και ακρίβεια.

Κλείνοντας, η ενασχόληση με την κατασκευή και την εφαρμογή των συγκεκριμένων αλγορίθμων προσέφερε πολύτιμες γνώσεις και εμπειρίες, αναδεικνύοντας την αξία της *Τεχνητής Νοημοσύνης* και του πεδίου της *Μηχανικής Μάθησης*.