



ΣΤΡΑΤΙΩΤΙΚΗ ΣΧΟΛΗ ΕΥΕΛΠΙΔΩΝ  
Τμήμα Στρατιωτικών Επιστημών

ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ  
ΔΙΔΡΥΜΑΤΙΚΟ ΔΙΑΤΜΗΜΑΤΙΚΟ  
ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ  
ΑΚΑΔΗΜΑΪΚΟΥ ΕΤΟΥΣ 2020-21  
Σχεδίαση και Επεξεργασία  
Συστημάτων (Systems Engineering)  
(ΠΔ 97 /2015/ΦΕΚ 163Α'/20.08.2014)



ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ  
Σχολή Μηχανικών Παραγωγής & Διοίκησης

# ΜΕΤΑΠΤΥΧΙΑΚΗ ΔΙΑΤΡΙΒΗ

## ΣΥΣΤΗΜΑ ΥΠΟΣΤΗΡΙΞΗΣ ΑΠΟΦΑΣΕΩΝ ΜΕ ΤΕΧΝΙΚΕΣ ΜΗΧΑΝΙΚΗΣ ΜΑΘΗΣΗΣ ΓΙΑ ΠΡΟΒΛΕΨΗ ΣΤΡΑΤΟΛΟΓΗΣΗΣ ΤΡΟΜΟΚΡΑΤΩΝ

ΠΑΠΑΔΟΠΟΥΛΟΣ ΕΜΜΑΝΟΥΗΛ

A.M.: 2016018015

ΑΘΗΝΑ, ΦΕΒΡΟΥΑΡΙΟΣ 2021



Η Μεταπτυχιακή Διατριβή του Παπαδόπουλου Εμμανουήλ εγκρίνεται:

**ΤΡΙΜΕΛΗΣ ΕΞΕΤΑΣΤΙΚΗ ΕΠΙΤΡΟΠΗ**

**Καραδήμας Νικόλαος** (Επιβλέπων),  
Επίκουρος Καθηγητής



.....

**Δάρας Νικόλαος,**  
Καθηγητής



**Τσαφάρakis Στέλιος,**  
Αναπληρωτής Καθηγητής

Παπαδόπουλος Εμμανουήλ



© Copyright  
Φεβρουάριος, 2021

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ' ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τους συγγραφείς



Οι αρχές και οι τεχνικές που αναφέρονται στην παρούσα εργασία εκφράζουν το συγγραφέα και δεν πρέπει να ερμηνευθεί ότι αντιπροσωπεύουν επίσημες θέσεις της Στρατιωτικής Σχολής Ευελπίδων ή του Πολυτεχνείου Κρήτης. Είναι, οι βέλτιστες διαθέσιμες κατά το χρόνο της συγγραφής αυτής της έρευνας.

*«Μπορούμε να πούμε ότι η Αναλυτική Μηχανή υφαίνει αλγεβρικά μοτίβα  
όπως και ο αργαλειός του Ζακλάρ που υφαίνει λουλούδια και φύλλα»*

*Augusta Ada Lovelace*

*(1815-1852)*



## ΕΥΧΑΡΙΣΤΙΕΣ

Η παρούσα διπλωματική εργασία πραγματοποιήθηκε στα πλαίσια του Διδρυματικού Διατμηματικού Μεταπτυχιακού προγράμματος «Σχεδίαση και Επεξεργασία Συστημάτων (Systems Engineering)» της Σχολής Ευελπίδων και της Σχολής Μηχανικών Παραγωγής και Διοίκησης του Πολυτεχνείου Κρήτης κατά τα έτη 2020-2021.

Την εποπτεία της εργασίας αυτής είχε ο Επίκουρος Καθηγητής Πληροφορικής της Σχολής Ευελπίδων Δρ. Καραδήμας Νικόλαος, τον οποίο και θέλω να ευχαριστήσω θερμά τόσο για την επιστημονική βοήθεια που μου παρείχε όσο και για την ευγενική υποστήριξή του και τις υποδείξεις κατά την εκπόνηση αυτής της εργασίας.





**ΑΚΡΩΝΥΜΙΑ**

TCP/IP	Transmission Control Protocol/Internet Protocol - Πρωτόκολλο Ελέγχου Μετάδοσης / Πρωτόκολλο Διαδικτύου
SMOTE	Synthetic Minority Oversampling Technique - Συνθετική υπερδειγματοληψία
KNN	k-nearest neighbors' algorithm - Αλγόριθμος Κ κοντινότεροι γείτονες
IR	Information Recovery - Ανάκτηση πληροφοριών
CSV	Comma-Separated Values - Αρχεία τιμών διαχωρισμένων με κόμματα
ISIS	Islamic State of Iraq and Syria – Ισλαμικό κράτος
PCA	Principal Component Analysis - Ανάλυση Κύριων Συνιστωσών
SVM	Support Vector Machine - Μηχανές Διανυσμάτων Υποστήριξης ΜΔΥ
SVC	Support Vector Classifier – Ταξινομητής Διανυσμάτων Υποστήριξης
SQL	Structured Query Language – Δομημένη γλώσσα αναζητήσεων
NLTK	Natural Language Toolkit - Εργαλείο Φυσικής Γλώσσας
TFIDF	Term Frequency – Inverse Document Frequency - Συχνότητα - Αντίστροφη Συχνότητα Εγγράφου



**ΠΕΡΙΕΧΟΜΕΝΑ**

<b>ΕΥΧΑΡΙΣΤΙΕΣ</b> .....	<b>7</b>
<b>ΑΚΡΩΝΥΜΙΑ</b> .....	<b>9</b>
<b>ΠΕΡΙΕΧΟΜΕΝΑ</b> .....	<b>11</b>
<b>ΠΕΡΙΛΗΨΗ</b> .....	<b>13</b>
<b>ABSTRACT</b> .....	<b>15</b>
<b>1 ΕΙΣΑΓΩΓΗ</b> .....	<b>17</b>
1.1 ΤΑ ΔΕΔΟΜΕΝΑ .....	19
<b>2 ΕΙΣΑΓΩΓΗ ΚΑΙ ΙΣΤΟΡΙΚΑ ΣΤΟΙΧΕΙΑ ΤΗΣ ΜΗΧΑΝΙΚΗΣ ΜΑΘΗΣΗΣ</b> .....	<b>23</b>
2.1 ΙΣΤΟΡΙΚΑ ΣΤΟΙΧΕΙΑ .....	24
2.1.1 Θεωρία Πιθανοτήτων .....	24
2.2 ΔΙΑΔΙΚΑΣΙΑ ΚΑΙ ΣΤΟΙΧΕΙΑ ΤΗΣ ΜΗΧΑΝΙΚΗΣ ΜΑΘΗΣΗΣ .....	25
2.3 ΜΗ ΕΠΟΠΤΕΥΟΜΕΝΗ ΜΑΘΗΣΗ (UNSUPERVISED LEARNING) .....	26
2.3.1 Ομαδοποίηση (Clustering) .....	27
2.3.2 Μείωση διαστάσεων .....	28
2.3.3 Γενετικά μοντέλα .....	28
2.3.4 Μη εποπτευόμενη βαθιά μάθηση .....	29
2.3.5 Παραδείγματα Μη Επιβλεπόμενης Μάθησης .....	29
2.4 ΕΠΙΒΛΕΠΟΜΕΝΗ ΜΑΘΗΣΗ (SUPERVISED LEARNING) .....	30
2.4.1 Ταξινόμηση (Classification) .....	31
2.4.2 Παλινδρόμηση (Regression) .....	32
2.4.3 Πως λειτουργεί ένας αλγόριθμος εποπτευόμενης μάθησης .....	32
2.4.4 Αλγόριθμοι επιβλεπόμενης μάθησης .....	32
2.4.5 Παραδείγματα Εποπτευόμενης Μάθησης .....	33
2.4.6 Αλγόριθμος Support Vector Machines .....	33
2.4.7 Bayes Αλγόριθμοι .....	37
2.4.8 Naïve Bayes .....	38
2.4.8.1 Παράδειγμα ενός Bayesian ταξινομητή .....	39
2.4.9 Δένδρα απόφασης (Decision trees) .....	40
2.4.10 Τυχαία Δάση (Random Forests) .....	42
2.5 ΕΠΕΞΕΡΓΑΣΙΑ ΦΥΣΙΚΟΥ ΛΟΓΟΥ (NATURAL LANGUAGE PROCESSING) .....	45
2.5.1 Δειγματοληψία Ενδείξεων (Tokenization) .....	46
2.5.2 Στελεχοποίηση (Stemming) .....	46
2.5.3 Λεμματοποίηση (Lemmatization) .....	49

2.6	ΜΗ ΙΣΟΡΡΟΠΗΜΕΝΑ ΔΕΔΟΜΕΝΑ ΚΑΙ ΑΝΤΙΜΕΤΩΠΙΣΗ .....	51
2.6.1	Συνθετική υπερδειγματοληψία - SMOTE (Synthetic Minority Oversampling Technique).....	52
2.7	ΕΞΑΓΩΓΗ ΧΑΡΑΚΤΗΡΙΣΤΙΚΩΝ ΛΕΚΤΙΚΩΝ ΔΕΔΟΜΕΝΩΝ.....	53
2.7.1	Σάκος Λέξεων - Bag of Words .....	54
2.7.2	TFIDF .....	54
2.7.3	TFIDF vs Bag of words .....	55
3	ΑΝΑΛΥΣΗ ΚΑΙ ΛΥΣΗ ΤΟΥ ΠΡΟΒΛΗΜΑΤΟΣ ΜΑΣ.....	57
3.1	ΔΙΑΔΙΚΑΣΙΑ ΠΡΟΕΠΕΞΕΡΓΑΣΙΑΣ ΛΕΚΤΙΚΩΝ ΔΕΔΟΜΕΝΩΝ .....	59
3.2	ΔΙΑΔΙΚΑΣΙΑ ΕΚΜΑΘΗΣΗΣ ΤΩΝ ΜΟΝΤΕΛΩΝ ΜΑΣ.....	63
4	ΑΠΟΤΕΛΕΣΜΑΤΑ .....	71
4.1	ΣΥΖΗΤΗΣΗ ΑΠΟΤΕΛΕΣΜΑΤΩΝ .....	71
4.2	ΤΡΟΠΟΙ ΕΠΕΚΤΑΣΗΣ.....	73
4.3	ΕΦΑΡΜΟΓΕΣ.....	74
	ΒΙΒΛΙΟΓΡΑΦΙΑ.....	75
	ΠΑΡΑΡΤΗΜΑ Α: ΚΩΔΙΚΑΣ.....	77

## ΠΕΡΙΛΗΨΗ

Στις μέρες μας λόγω των πολλαπλών τρομοκρατικών επιθέσεων, η ανάγκη αντιμετώπισης τους ή ακόμα καλύτερα η ανάγκη πρόβλεψής τους, γίνεται όλο και μεγαλύτερη. Η συνεχόμενα εξελισσόμενη τεχνολογία και οι δυνατότητες επεξεργασίας δεδομένων και προτυποποίησης που έχουμε διαθέσιμες, μας δίνουν την ευκαιρία να χρησιμοποιήσουμε αυτές τις τεχνικές, ώστε να μπορέσουμε να δημιουργήσουμε κάτι χρήσιμο για την αντιμετώπιση του προαναφερθέντος προβλήματος.

Στην παρούσα μεταπτυχιακή εργασία θα αναλυθεί και θα γίνει μία αρχική επικύρωση της ιδέας (proof of concept) για την αναγνώριση λεκτικών προτύπων από αναρτήσεις ιστολογίου (blog posts) και φόρουμ του σκοτεινού διαδικτύου (dark web), ώστε να βρεθεί η συσχέτιση των λέξεων ως προς το αν αυτά απευθύνονται σε τρομοκρατική στρατολόγηση ή όχι.

Η εργασία έχει χωριστεί σε τρεις κύριες ενότητες, οι οποίες περιλαμβάνουν την ανάλυση των λεκτικών δεδομένων και χρήση του κατάλληλου τρόπου επεξεργασίας τους, ώστε να εισέλθουν σε μοντέλο μηχανικής Μάθησης, την χρήση πολλαπλών μοντέλων μηχανικής μάθησης, ώστε να βρεθούν τα καλύτερα ως προς την απόδοση πρόβλεψης, καθώς και συζήτηση και μελλοντικοί τρόποι επέκτασης της εργασίας.



## ABSTRACT

Nowadays, due to the multiple terrorist attacks, the need to deal with them or even better the need to anticipate them, is becoming more imperative. The constantly evolving technology and the data processing and standardization capabilities that we have available, give us the opportunity to use these techniques, so that we can create something useful to address the aforementioned problem.

This master thesis will be a proof of concept for the recognition of verbal patterns from dark web forums blog posts in order to find the correlation of words in terms of whether they are targeted to terrorist recruitment or not.

The work is divided into three main sections, which include the analysis of verbal data and the use of the appropriate way of processing them to enter a machine learning model, the use of different machine learning models, in order to find the best in terms of predictive performance, as well as discussion and ways of extending the work in the future.





## 1 ΕΙΣΑΓΩΓΗ

Είναι γεγονός ότι στις μέρες μας, οι νέες τεχνολογίες έχουν οδηγήσει στην βελτίωση των συνθηκών διαβίωσης της κοινωνίας μας. Οι άνθρωποι έχουν προσαρμόσει την επαγγελματική και την προσωπική τους ζωή στις σύγχρονες αυτές συνθήκες. Το κύριο χαρακτηριστικό της σημερινής εποχής, είναι η ροή πληροφορίας και οι τρόποι που αυτή διαχέεται. Δείγμα αυτού είναι ο χαρακτηρισμός της εποχής μας ως εποχή της πληροφορίας.

Στην εποχή της πληροφορίας, το πιο χαρακτηριστικό φαινόμενο είναι η συνεχώς αυξανόμενη ενασχόληση των ανθρώπων με τα μέσα κοινωνικής δικτύωσης – social media. Ειδικότερα, φαίνεται να αυξάνεται η εξάρτηση του πληθυσμού από αυτή την καινούρια μορφή πληροφόρησης, είτε για απλά θέματα της καθημερινότητας, είτε για παγκόσμιας κλίμακας αντικείμενα. Το φαινόμενο ενασχόλησης με τα μέσα κοινωνικής δικτύωσης, πέρα του να ενημερωθεί κάποιος για την τρέχουσα επικαιρότητα, είναι η «μοντέρνα κοινωνικοποίηση». Ζούμε σε μια εποχή όπου επιστήμονες διαφορετικών ειδικοτήτων καλούνται να ερμηνεύσουν τον αντίκτυπο και τα αποτελέσματα της χρήσης του Διαδικτύου στην καθημερινή ζωή. Τα πληροφοριακά συστήματα αναπτύσσονται και επεκτείνονται με αστραπιαία ταχύτητα. Καθημερινά, εκατομμύρια χρηστών, έχουν τη δυνατότητα να αλληλοεπιδρούν, να επικοινωνούν, να εκφράζουν και να δημοσιεύουν τις απόψεις τους. Αυτό συμβαίνει, μέσω της δυνατότητας που έχουν να δημιουργήσουν διάφορους ιστοχώρους - blogs, ή μέσω της συμμετοχής τους σε διάφορα είδη αντίστοιχων εφαρμογών και κοινωνικών δικτύων.

Στην εποχή της πληροφορίας, η τρομοκρατία αποτελεί μία ξεχωριστή περίπτωση δημόσιου ενδιαφέροντος, για την οποία εκφράζονται τοποθετήσεις αλλά και γίνεται ανταλλαγή πληροφοριών, από το σύνολο των σύγχρονων μέσων κοινωνικής δικτύωσης.

Το κύριο μέσο ροής πληροφοριών είναι το Διαδίκτυο και μέσω αυτού δημιουργούνται τα κοινωνικά δίκτυα και η επικοινωνία διαχέεται από τα μέσα κοινωνικής δικτύωσης κυρίως, που θα παρουσιαστούν παρακάτω. Αρχικά, θα περιγραφεί και θα οριστεί τι είναι το διαδίκτυο. Το Διαδίκτυο (Internet) (σύμφωνα με την ηλεκτρονική εγκυκλοπαίδεια Wikipedia) είναι ένα «παγκόσμιο σύστημα διασυνδεδεμένων δικτύων υπολογιστών, οι οποίοι χρησιμοποιούν καθιερωμένη ομάδα πρωτοκόλλων, η οποία συχνά αποκαλείται "TCP/IP" (αν και αυτή δεν χρησιμοποιείται από όλες τις υπηρεσίες του Διαδικτύου) για να εξυπηρετεί δισεκατομμύρια χρήστες καθημερινά σε ολόκληρο τον κόσμο. Οι διασυνδεδεμένοι ηλεκτρονικοί υπολογιστές ανά τον κόσμο, οι οποίοι βρίσκονται σε ένα κοινό δίκτυο επικοινωνίας, ανταλλάσσουν μηνύματα (πακέτα) με τη χρήση διαφόρων πρωτοκόλλων (τυποποιημένοι κανόνες επικοινωνίας), τα οποία υλοποιούνται σε επίπεδο υλικού και λογισμικού. Το κοινό αυτό δίκτυο καλείται Διαδίκτυο.».

Τα κύρια χαρακτηριστικά του είναι:

- ότι είναι μέσο αμφίδρομο, δηλαδή και παρέχει δεδομένα – πληροφορίες προς τον χρήστη και ο χρήστης μπορεί να τροφοδοτήσει με δεδομένα και πληροφορίες προς την αντίθετη κατεύθυνση, δηλαδή προς το Διαδίκτυο και τα μέσα προβολής που αυτό διαθέτει.
- άμεσο, με την έννοια της αστραπιαίας μετάδοσης της πληροφορίας, μέσω των δικτύων δεδομένων παγκοσμίως, από την στιγμή που μία συσκευή που παρέχει την δυνατότητα σύνδεσης με αυτό, μπορεί να διανείμει και να δημοσιοποιήσει περιεχόμενο.
- διαχέει περιεχόμενο, δηλαδή τα δεδομένα – οι πληροφορίες, έχουν την δυνατότητα διάχυσης προς άπειρες κατευθύνσεις.

Παρατηρείται στις μέρες μας, στα διάφορα πεδία συγκρούσεων και ειδικότερα στις τρομοκρατικές επιθέσεις και την στρατολόγηση τρομοκρατών που αφορούν την παρούσα εργασία, ότι το πεδίο ενδιαφέροντος εκτείνεται πλέον σε όλο το εύρος του πληθυσμού και της κοινωνίας. Επιπροσθέτως, ο έλεγχος της ροής της πληροφορίας είναι δύσκολος, εφόσον οποιοσδήποτε μπορεί με μία απλή συσκευή να δημιουργήσει ή να αναπαράξει δεδομένα και να τα διανείμει προς «άπειρες» κατευθύνσεις, παγκοσμίως.

Επιπλέον, η νέα αυτή μορφή επικοινωνίας, προσφέρει ένα εξαιρετικό μέσο για τη διακίνηση πληροφοριών για εξτρεμιστικές ομάδες ή οπαδούς τέτοιων ακραίων ομάδων, ώστε να χτίσουν την αυτονομία τους και να οργανωθούν - συντονιστούν. Οι δραστηριότητες των ομάδων αυτών όπως για παράδειγμα: η Χεζμπολάχ (Hezbollah), οι Ταλιμπάν (Taliban), η Αλ Κάιντα (AlQaeda) και άλλες, απομακρύνονται από τις κλασικές τακτικές πολέμου και χαρακτηρίζονται πλέον από εκτενή παρουσία εντός του διαδικτύου. Οι τακτικές αυτές, εντάσσονται σε ένα μεγαλύτερο σύνολο αλλαγών, το οποίο πραγματοποιήθηκε στις σύγχρονες κοινωνίες ως συνέπεια των νέων τεχνολογικών εξελίξεων. Παρατηρείται ότι τα περισσότερα χτυπήματα – επιθέσεις από εξτρεμιστικές - τρομοκρατικές οργανώσεις και άλλες περιθωριακές και μειονοτικές ομάδες, λαμβάνουν χώρα στα πλαίσια του αστικού ιστού και κυρίως σε μεγάλες συναθροίσεις πληθυσμών και κατοίκων, με σκοπό τη μεγιστοποίηση του φονικού και του τρομοκρατικού αποτελέσματος, όπως και της μετάδοσης του ψυχολογικού κλίματος που θέλουν να επιτύχουν οι ομάδες αυτές.

Στην εποχή της πληροφορίας, η τρομοκρατία είναι ιδιαίτερα εξελιγμένη ως προς τις μεθόδους και την τεχνολογία που χρησιμοποιεί. Η προπαγάνδα, η στρατολόγηση και η οργάνωση - συντονισμός μέσω του Διαδικτύου είναι το σημαντικότερο όπλο της νέας εποχής της τρομοκρατίας, κάτι το οποίο αποτελεί και το έναυσμα αυτής της εργασίας.

Οι τρομοκράτες έχουν ήδη αναγνωρίσει από την αρχή του 21ου αιώνα και χρησιμοποιούν αυτή τη νέα τεχνολογία εκμεταλλευόμενοι τις δυνατότητες που δίνει το σκοτεινό Διαδίκτυο (Dark Web), αλλά και αυτές των γνωστών κοινωνικών δικτύων και πλατφορμών όπως το Facebook, το Instagram, το Twitter και το

YouTube. Άγνωστος είναι ο ακριβής αριθμός των μελών των τρομοκρατικών οργανώσεων που δρουν παγκοσμίως. Καθημερινά, φαίνεται ότι αυξάνονται οι υποστηρικτές της τρομοκρατίας, κυρίως μεταξύ των φτωχών στρωμάτων της κοινωνίας. Ο κόσμος είναι κατά ένα μεγάλο ποσοστό συνδεδεμένος (online) καθ' όλη τη διάρκεια της ημέρας και για πολλούς διαφορετικούς λόγους, όπως αναφέρθηκε παραπάνω. Αυτό το παγκόσμιο δίκτυο που χρησιμεύει ως μία πλατφόρμα ανταλλαγής ειδήσεων και απόψεων είναι ένα πανίσχυρο εργαλείο για τη διεθνή τρομοκρατία. Χρησιμοποιώντας την ισχύ του Διαδικτύου, οι τρομοκρατικές ομάδες προσπαθούν να εκμεταλλευτούν τα μεγάλης κλίμακας αθλητικά ή πολιτιστικά γεγονότα, δίνοντας τους την δύναμη να διαδώσουν την ιδεολογία τους και να στρατολογήσουν νέους τρομοκράτες ή να υποκινήσουν υφιστάμενους απομονωμένους εξτρεμιστικούς πυρήνες, κάτι που έχει κάνει συστηματικά το ονομαζόμενο Ισλαμικό Κράτος (ISIS).

Το σκοτεινό διαδίκτυο (Dark Web), αλλά ακόμα και τα γνωστά κοινωνικά δίκτυα και πλατφόρμες χρησιμοποιούνται επίσης από τους «εγκεφάλους» - ηγετικά στελέχη τρομοκρατικών οργανώσεων ώστε να οργανώσουν και να δώσουν εντολή για την επίθεσή τους.

Με το θέμα των social media και τον ρόλο τους στις σύγχρονες στρατιωτικές επιχειρήσεις έχει ασχοληθεί η Ιωάννα Ηλιάδη, υποψήφια διδάκτορας του Ανοικτού Πανεπιστημίου Κύπρου πάνω στην Επικοινωνία και τη Δημοσιογραφία, η οποία αναφέρει ότι: «Οι αλλαγές στον τομέα της επικοινωνίας και πιο συγκεκριμένα η δυνατότητα που έχουμε για φορητό ίντερνετ, επιφέρουν δομικές αλλαγές στην κοινωνία μας. Τα social media χαρακτηρίζονται από το χάος και την πολυπλοκότητα, ενώ η βιομηχανική εποχή την οποία αφήνουμε πίσω μας βασίζεται στην ιεραρχία και την τυποποίηση» αναφέρει σχετικά, τονίζοντας ότι «οι τακτικές του παρελθόντος δεν ταιριάζουν με το νέο μοντέλο πολέμου».

Ένα σημαντικό ζήτημα είναι η προσαρμοστικότητα των οργανώσεων αυτών. Οι τρομοκράτες φαίνεται να αναζητούν νέες, λιγότερο γνωστές πλατφόρμες κοινωνικής δικτύωσης για προπαγανδιστικούς σκοπούς, εξαιτίας της συντονισμένης αντίδρασης εναντίον τους στα γνωστά κοινωνικά δίκτυα. Η παρούσα εργασία επικεντρώνεται στην δραστηριότητα στρατολόγησης τρομοκρατών στο σκοτεινό Διαδίκτυο (Dark Web).

## 1.1 Τα δεδομένα

Τα δεδομένα που χρησιμοποιήθηκαν στην παρούσα εργασία συλλέχθηκαν από φόρουμ από το σκοτεινό Διαδίκτυο (Dark Web). Τα φόρουμ οργανώνονται γλωσσικά κυρίως σε Αγγλικά, Αραβικά, Γαλλικά, Γερμανικά και Ρωσικά. Δεδομένα από είκοσι οκτώ φόρουμ συλλέχθηκαν έως το 2012 από το Εργαστήριο Τεχνητής Νοημοσύνης για να υποστηρίξουν το σχέδιο Dark Web για τη μελέτη των διεθνών κοινωνικών μέσων και κινήσεων της Τζιχάντ. Κάθε συλλογή περιέχει έως και εκατομμύρια δημοσιεύσεις που έχουν γραφτεί από χιλιάδες μέλη του φόρουμ. Οι καταχωρήσεις οργανώνονται σε νήματα (threads) που γενικά

υποδεικνύουν το υπό συζήτηση θέμα. Κάθε δημοσίευση περιλαμβάνει λεπτομερή μεταδεδομένα (metadata), όπως ημερομηνία, όνομα μέλους κ.α. Το δείγμα δεδομένων του φόρουμ που έχουμε, παρέχεται ως αρχείο συμπιεσμένου κειμένου που μπορεί να μεταφορτωθεί και στη συνέχεια να ανοίξει σε οποιοδήποτε πρόγραμμα επεξεργασίας κειμένου συμβατό με αρχεία τιμών διαχωρισμένων με κόμματα (Comma-Separated Values - CSV)<sup>1</sup>.

Τα βαθύτερα στρώματα του Deep Web, ενός τμήματος γνωστού ως Dark Net, περιέχουν δεδομένα που έχουν αποκρυφτεί σκόπιμα, συμπεριλαμβανομένων των παράνομων και αντικοινωνικών πληροφοριών. Το Dark Net μπορεί να οριστεί ως το τμήμα του Deep Web που μπορεί να προσεγγιστεί μόνο μέσω ειδικών προγραμμάτων περιήγησης (όπως το πρόγραμμα περιήγησης Tor). Οι μελετητές Daniel Moore and Thomas Rid από το King's College του Λονδίνου διαπίστωσαν ότι το 57% του Dark Net περιέχει παράνομο περιεχόμενο όπως παράνομη πορνογραφία, παράνομες οικονομικές συναλλαγές, διακίνηση ναρκωτικών, παράνομη εμπορία όπλων, διακίνηση πλαστών νομισμάτων και πολλά άλλα. Το Dark Net έχει συνδεθεί με το περίφημο WikiLeaks, καθώς και το Bitcoin, που λέγεται ότι είναι το νόμισμα του Dark Net.

Οι τρομοκράτες δραστηριοποιούνται σε διάφορες ηλεκτρονικές πλατφόρμες από τα τέλη της δεκαετίας του 1990. Το Surface Web ωστόσο, φάνηκε ότι ήταν πολύ επικίνδυνο για τους τρομοκράτες που αναζητούσαν ανωνυμία: θα μπορούσαν να παρακολουθούνται, να εντοπίζονται και να βρίσκονται. Πολλές από τις ιστοσελίδες που σχετίζονται με τρομοκρατία, καθώς και τα μέσα κοινωνικής δικτύωσης και ενημέρωσης που βρίσκονται στο Surface Web παρακολουθούνται από αντιτρομοκρατικά γραφεία τα οποία μπορούν να παρέμβουν. Αντίθετα, στο δίκτυο Dark Net, τα αποκεντρωμένα και ανώνυμα δίκτυα βοηθούν στην αποφυγή της σύλληψης ατόμων και του κλεισίματος πλατφορμών που διακινούν δεδομένα σχετικά με τρομοκρατία. Δραστηριότητες του ISIS στο Surface Web παρακολουθούνται στενά και η απόφαση από έναν αριθμό κυβερνήσεων ή και οργανισμών να καταργήσουν - φιλτράρουν το εξτρεμιστικό περιεχόμενο έχουν ωθήσει τους τρομοκράτες να αναζητήσουν νέους τρόπους επικοινωνίας.

Τα μέσα κοινωνικής δικτύωσης έχουν περίπλοκο ιστορικό με την τρομοκρατία. Καθώς οργανώσεις όπως το Ισλαμικό Κράτος αξιοποιούσαν τη δύναμη των κοινωνικών πλατφορμών για στρατολόγηση, κάλεσμα για δράση και προπαγάνδα, τέθηκαν αρχικά θέματα δικαιωμάτων ελευθερίας λόγου των τρομοκρατών.

Μετά τη συντριπτική δημόσια πίεση και τις απειλές της νέας νομοθεσίας σε ολόκληρο τον κόσμο, τα μέσα κοινωνικής δικτύωσης αναγκάστηκαν να αρχίσουν να ασχολούνται πιο ενεργά με τις αντιτρομοκρατικές προσπάθειες. Ωστόσο, οι προσπάθειες που γίνονται για την καταπολέμηση της τρομοκρατίας απέχουν πολύ από το να είναι όσο επιτυχείς παρουσιάζονται

<sup>1</sup> Τα αποθηκευμένα αρχεία τύπου CSV αναφέρονται σε αρχεία δεδομένων που επισυνάπτεται με την .csv επέκταση, και είναι αρχεία τιμών διαχωρισμένα με κόμματα.

Τα μέσα κοινωνικής δικτύωσης έχουν δηλώσει ότι καταβάλλουν προσπάθειες ώστε να χρησιμοποιήσουν τη μηχανική μάθηση και μαύρες λίστες περιεχομένου για να αποκλείσουν το τρομοκρατικό περιεχόμενο από τις πλατφόρμες τους. Μέχρι σήμερα δεν υπάρχει βεβαιότητα για το αν οι αλγόριθμοι μηχανικής μάθησης των μέσων κοινωνικής δικτύωσης έχουν φτάσει στο σημείο να είναι πράγματι χρήσιμοι ή αν βασίζονται αποκλειστικά στις μαύρες λίστες περιεχομένου τους για να αφαιρέσουν υλικό σχετικό με τρομοκρατία.

Στην παρούσα εργασία θα εξεταστεί η δυνατότητα χρήσης και θα εξακριβωθεί η αποτελεσματικότητα των αλγορίθμων μηχανικής μάθησης στο πεδίο ανίχνευσης στρατολόγησης τρομοκρατών και πρόβλεψης αντίστοιχων χτυπημάτων.

Στο κεφάλαιο 2, γίνεται μία εισαγωγή στην μηχανική μάθηση καθώς και στα ιστορικά στοιχεία. Επίσης παρουσιάζονται η διαδικασία και τα στοιχεία της μηχανικής μάθησης και οι αλγόριθμοι που θα εξεταστούν. Στη συνέχεια εξετάζονται οι μηχανισμοί επεξεργασίας φυσικού λόγου και αναλύεται το πρόβλημα των μη ισορροπημένων δεδομένων και οι τρόποι αντιμετώπισής του. Το κεφάλαιο κλείνει με την ανάλυση των μηχανισμών εξαγωγής των χαρακτηριστικών λεκτικών δεδομένων. Στη συνέχεια, στο κεφάλαιο 3 παρουσιάζεται η ανάλυση και λύση του προβλήματός μας, ξεκινώντας από την διαδικασία προ επεξεργασίας των λεκτικών δεδομένων, την διαδικασία εκμάθησης των μοντέλων μας, και καταλήγοντας στην εξαγωγή των αποτελεσμάτων. Τέλος στο κεφάλαιο 4 γίνεται η αξιολόγηση των αποτελεσμάτων και αναφορά στους τρόπους επέκτασης της παρούσας εργασίας και σχετικές εφαρμογές.



## 2 ΕΙΣΑΓΩΓΗ ΚΑΙ ΙΣΤΟΡΙΚΑ ΣΤΟΙΧΕΙΑ ΤΗΣ ΜΗΧΑΝΙΚΗΣ ΜΑΘΗΣΗΣ.

Η μηχανική μάθηση είναι αναμφισβήτητα μία από τις πιο ισχυρές τεχνολογίες, με μεγάλη επιρροή στον σημερινό κόσμο και η συνεχής εξέλιξη και ενσωμάτωση της σε όλο και περισσότερους τομείς της ανθρώπινης δραστηριότητας, δείχνουν ότι δεν έχει απελευθερώσει ακόμα το πλήρες δυναμικό της.

Η μηχανική μάθηση είναι ένα χρήσιμο εργαλείο για τη μετατροπή των πληροφοριών σε γνώση, ειδικά τα τελευταία χρόνια που έχει σημειωθεί μία έκρηξη στον όγκο των δεδομένων. Αυτός ο τεράστιος όγκος δεδομένων είναι άχρηστος αν δεν αναλυθεί ώστε να βρεθούν τα πρότυπα που είναι κρυμμένα μέσα σε αυτόν. Οι τεχνικές μηχανικής μάθησης χρησιμοποιούνται για την αυτόματη εύρεση των πολύτιμων υποκείμενων μοτίβων (η προτύπων) μέσα σε πολύπλοκα δεδομένα που διαφορετικά θα ήταν μάλλον απίθανο να ανακαλύψουμε. Τα κρυμμένα μοτίβα και η γνώση ενός προβλήματος μπορούν να χρησιμοποιηθούν για να προβλέψουν μελλοντικά γεγονότα και να εκτελέσουν κάθε είδους πολύπλοκης διαδικασίας λήψης αποφάσεων.

Οι περισσότεροι από εμάς δεν γνωρίζουν ότι ήδη υπάρχει καθημερινή αλληλεπίδραση με την Μηχανική Μάθηση. Κάθε φορά που εκτελείται μία αναζήτηση πληροφορίας υπό τη μορφή κειμένου για να βρεθεί μια πληροφορία ή η αναζήτηση φωτογραφιών στις γνωστές μηχανές αναζήτησης, ακόμα και η διαδικασία εκτέλεσης ενός αρχείου μουσικής και η αυτόματη αναζήτηση των μεταδεδομένων του, είναι μέρος ενός κρυφού ευφυούς μηχανισμού μηχανικής μάθησης, ο οποίος με κάθε αλληλεπίδραση γίνεται ακόμα αποδοτικότερος. Η μηχανική μάθηση είναι επίσης πίσω από τις εξελίξεις που αλλάζουν στον κόσμο, όπως η ανίχνευση του καρκίνου, η δημιουργία νέων φαρμάκων και η αυτόνομη οδήγηση των αυτοκινήτων. Ο λόγος που η μηχανική μάθηση είναι τόσο συναρπαστική είναι επειδή προχωρά ένα βήμα παραπέρα από όλα τα προηγούμενα συστήματα βασισμένα σε απλούς κανόνες τύπου:

Εάν (Συνθηκη)  $\rightarrow$  Κάνε  $z$

Π.χ. if( $x = y$ ): do  $z$

Παραδοσιακά, η τεχνολογία λογισμικού συνδύαζε ανθρώπους που δημιουργούσαν κανόνες, μαζί με δεδομένα, για να δημιουργήσουν απαντήσεις σε ένα πρόβλημα. Αντ' αυτού, η μηχανική μάθηση χρησιμοποιεί δεδομένα και απαντήσεις για να ανακαλύψει τους κανόνες πίσω από ένα πρόβλημα. Για να μάθουν τους κανόνες που διέπουν ένα φαινόμενο, οι μηχανές πρέπει να περάσουν από μια διαδικασία μάθησης, να δοκιμάσουν διαφορετικούς κανόνες και να μάθουν από το πόσο αποτελεσματικά εκτελούν ώστε να φτάσουν στο επιθυμητό αποτέλεσμα. Αυτή η διαδικασία μάθησης των κανόνων που διέπουν ένα φαινόμενο, είναι γνωστή ως Μηχανική Μάθηση (Machine Learning).



Υπάρχουν πολλές μορφές μηχανικής μάθησης. Εποπτευόμενη (Supervised Learning), μη εποπτευόμενη (Unsupervised Learning), ημι-εποπτευόμενη (semi-supervised Learning) και ενισχυτική μάθηση (Reinforcement Learning). Κάθε μορφή μηχανικής μάθησης έχει διαφορετικές προσεγγίσεις, αλλά όλες ακολουθούν την ίδια υποκείμενη διαδικασία και θεωρία. Παρακάτω θα γίνει αναφορά στα στοιχεία αλλά και την διαδικασία που ακολουθείται κατά την εκμάθηση ενός μοντέλου.

## 2.1 Ιστορικά στοιχεία

Η Augusta Ada Lovelace (Άντα Λάβλεϊς) 1815 - 1852, κόρη του Λόρδου Byron και της Annabella Milbanke ήταν μια από τις ιδρύτριες προσωπικότητες της πληροφορικής και κατά τους ιστορικούς αυτή που δημιούργησε το πρώτο πρόγραμμα υπολογιστή. Η Lovelace προέβλεψε ότι οι μηχανές στο εγγύς μέλλον θα μπορούσαν όχι μόνο να επιλύουν μαθηματικά προβλήματα, αλλά και να συνθέτουν πολύπλοκη μουσική και να παράγουν γραφικά. Αυτό σήμαινε ότι θα μπορούσαν να δημιουργηθούν μαθηματικοί τύποι που θα μπορούσαν να αντλήσουν τις σχέσεις που αντιπροσωπεύουν οποιοδήποτε φαινόμενο. Η Lovelace συνειδητοποίησε ότι οι μηχανές θα μπορούσαν να αποκτήσουν τη δυνατότητα να κατανοήσουν τον κόσμο χωρίς την ανάγκη ανθρώπινης βοήθειας. Αν και η ζωή της ήταν σύντομη (έζησε μόνο 37 έτη), το έργο της Augusta Ada Lovelace, συμμετείχε για περισσότερο από ένα αιώνα σε αυτό που ονομάζεται «σύγχρονη επιστήμη υπολογιστών». Η συνεργασία της με τον Charles Babbage και τις Υπολογιστικές του Μηχανές ήταν καθοριστική για την εξέλιξη της σύγχρονης επιστήμης των υπολογιστών. Σχεδόν 200 χρόνια αργότερα, αυτές οι ιδέες είναι θεμελιώδεις στη Μηχανική Μάθηση. Ανεξάρτητα από το ποιο είναι το πρόβλημα, οι πληροφορίες μπορούν να γραφτούν σε ένα γράφημα ως σημεία δεδομένων. Η Μηχανική Μάθηση στη συνέχεια προσπαθεί να βρει τα μαθηματικά πρότυπα και τις σχέσεις που κρύβονται μέσα στις αρχικές πληροφορίες.

### 2.1.1 Θεωρία Πιθανοτήτων

Ένας άλλος μαθηματικός, ο Thomas Bayes (Τόμας Μπέϋς) 1701 - 1761, θεμελίωσε ιδέες που είναι απαραίτητες στη θεωρία των πιθανοτήτων που υπεισέρχονται στη μηχανική μάθηση.

Ζούμε σε έναν πιθανοτικό κόσμο. Όλα όσα συμβαίνουν έχουν την αβεβαιότητα που συνδέεται με αυτά. Η Bayesian ερμηνεία της πιθανότητας είναι η βάση της Μηχανικής Μάθησης. Στην ερμηνεία κατά Bayes, η πιθανότητα μετρά τον βαθμό αλήθειας. Το θεώρημα του Bayes τότε συνδέει το βαθμό αλήθειας σε μια πρόταση πριν και μετά τον υπολογισμό των δεδομένων. Εξαιτίας αυτού, οι πιθανότητες μας θα πρέπει να βασιστούν στις διαθέσιμες πληροφορίες σχετικά με ένα γεγονός, αντί στον αριθμό και τα αποτελέσματα των επαναλαμβανόμενων δοκιμών. Για παράδειγμα, στην πρόβλεψη του αποτελέσματος ενός ποδοσφαιρικού



αγώνα, αντί να βασιστεί η πρόβλεψη στο πλήθος των φορών που μία ομάδα έχει κερδίσει εναντίον μιας άλλης, μια Bayesian προσέγγιση θα χρησιμοποιούσε σχετικές πληροφορίες όπως η τρέχουσα φόρμα των παιχτών, τη θέση της ομάδας στο πρωτάθλημα και άλλα.

Το μειονέκτημα της λήψης αποφάσεων μέσα από αυτή την προσέγγιση είναι ότι οι πιθανότητες μπορούν ακόμη να ανατεθούν σε σπάνια γεγονότα, καθώς η διαδικασία λήψης αποφάσεων βασίζεται σε συναφή χαρακτηριστικά και συλλογιστική.

## 2.2 Διαδικασία και στοιχεία της Μηχανικής Μάθησης.

Για κάθε πρόβλημα που καλείται να επιλυθεί με τεχνικές μηχανικής μάθησης είναι απαραίτητα τα παρακάτω.

**Σετ δεδομένων:** Ένα σύνολο παραδειγμάτων δεδομένων που περιέχουν στοιχεία σημαντικά για την επίλυση του προβλήματος.

**Χαρακτηριστικά:** Σημαντικά κομμάτια δεδομένων που μας βοηθούν στην κατανόηση του προβλήματος. Αυτά τροφοδοτούνται σε έναν αλγόριθμο Machine Learning για να τον βοηθήσουν να μάθει μια σχέση ανάμεσα στα χαρακτηριστικά και τα αποτελέσματα προς πρόβλεψη.

**Μοντέλο:** Η παράσταση (εσωτερικό μοντέλο) ενός φαινομένου που έχει μάθει ένας αλγόριθμος Machine Learning. Το μοντέλο μαθαίνει από τα δεδομένα που εμφανίζονται κατά την εκπαίδευση. Το μοντέλο είναι η έξοδος που λαμβάνεται μετά την κατάρτιση ενός αλγορίθμου. Για παράδειγμα, ένας αλγόριθμος δέντρων αποφάσεων θα εκπαιδευτεί και θα παράγει ένα μοντέλο δέντρων αποφάσεων.

Σε κάθε πρόβλημα το οποίο λύνεται μέσω μηχανικής μάθησης ακολουθείται πάντα η παρακάτω διαδικασία.

**Συλλογή δεδομένων:** Η αρχική συλλογή όλων των δεδομένων τα οποία βοηθάνε ώστε να λυθεί το πρόβλημα που θέλουμε να λύσουμε.

**Προετοιμασία δεδομένων:** Μορφοποίηση και επεξεργασία των δεδομένων στη βέλτιστη μορφή τους, εξαγωγή των σημαντικών χαρακτηριστικών και κάποιες φορές μείωση διαστάσεων (dimensionality reduction).

**Επιλογή μοντέλου:** Επιλογή του μοντέλου μηχανικής μάθησης που θα χρησιμοποιηθεί, ανάλογα με τη φύση του προβλήματος.

**Εκπαίδευση:** Επίσης γνωστό ως στάδιο προσαρμογής (fit). Σε αυτό το στάδιο ο αλγόριθμος Machine Learning μαθαίνει την μαθηματική σχέση (η συνάρτηση) που διέπει τα δεδομένα.

**Αξιολόγηση:** Δοκιμάζεται το μοντέλο σε δεδομένα που δεν έχει ξαναδεί (validation η dev set) για να αξιολογηθεί το πόσο καλά εκτελείται.

**Προσαρμογή:** Προσαρμογή του μοντέλου για να μεγιστοποιηθεί η απόδοσή του.



Εικόνα 2-1. Διαδικασία λύσης ενός προβλήματος Μηχανικής Μάθησης.

Υπάρχουν πολλές προσεγγίσεις που μπορούν να ακολουθηθούν κατά τη διεξαγωγή της Μηχανικής Μάθησης. Συνήθως ομαδοποιούνται στις κατηγορίες που αναφέρονται παρακάτω.

Οι εποπτευόμενες (supervised) και οι μη εποπτευόμενες (unsupervised) είναι καλά καθιερωμένες προσεγγίσεις και οι πιο συνηθισμένες μέθοδοι.

Η ημι-εποπτευόμενη και η ενίσχυση της μάθησης είναι νεότερες και πιο περίπλοκες αλλά έχουν δείξει εντυπωσιακά αποτελέσματα.

Το θεώρημα No Free Lunch είναι γνωστό στη μηχανική μάθηση. Δηλώνει ότι δεν υπάρχει ενιαίος αλγόριθμος που θα λειτουργεί καλά για όλες τις εργασίες. Κάθε πρόβλημα πρέπει να προσεγγιστεί με τις ιδιαιτερότητές του. Ως εκ τούτου, υπάρχουν πολλοί αλγόριθμοι και προσεγγίσεις για να ταιριάζουν σε κάθε πρόβλημα αλλά και μεμονωμένες ιδιορρυθμίες.

Οι τάσεις δείχνουν ότι όλο και περισσότερες παραλλαγές τεχνικών Machine Learning και Artificial Intelligence (AI) θα συνεχίσουν να εμφανίζονται, που να ταιριάζουν καλύτερα σε διαφορετικούς τύπους προβλημάτων.

## 2.3 Μη Εποπτευόμενη Μάθηση (Unsupervised Learning)

Στην μάθηση χωρίς επίβλεψη παρέχονται μόνο τα δεδομένα εισόδου στα παραδείγματα. Δεν υπάρχουν επισημασμένα παραδείγματα εξόδων για να επιδιώξουμε. Ωστόσο, είναι σημαντικό να αναφερθεί ότι είναι δυνατό να βρεθούν πολλά ενδιαφέροντα και σύνθετα μοτίβα κρυμμένα μέσα στα δεδομένα χωρίς καμία ετικέτα (label).

Ένα παράδειγμα μάθησης χωρίς επίβλεψη στην πραγματική ζωή θα μπορούσε να είναι η ταξινόμηση διαφορετικών χαρτονομισμάτων σε χωριστούς σωρούς. Ενώ δεν έχει γίνει κάποια εκπαίδευση για τον

διαχωρισμό, με την εξέταση κάποιων χαρακτηριστικών τους όπως το χρώμα, μπορεί να βρεθεί το πως αυτά συσχετίζονται και να ταξινομηθούν στις σωστές τους ομάδες.

Η μη εποπτευόμενη μάθηση μπορεί να είναι πιο δύσκολη ως προς την κατανόηση των αποτελεσμάτων από την εποπτευόμενη μάθηση, καθώς η αφαίρεση της εποπτείας δηλώνει ότι το πρόβλημα έχει αναγνωριστεί - επιβεβαιωθεί σε μικρότερο βαθμό.

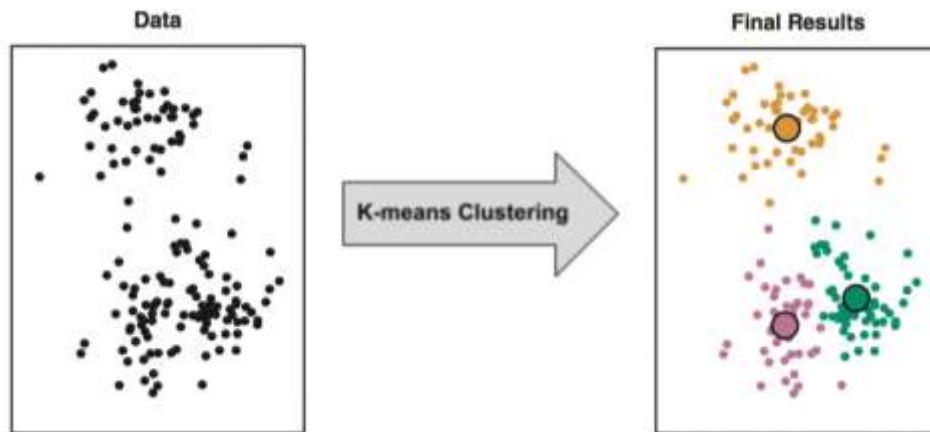
Ο αλγόριθμος έχει μια λιγότερο επικεντρωμένη ιδέα για τα πρότυπα που πρέπει να αναζητήσουμε. Εδώ μπορεί να δοθεί ένα παράδειγμα σχετικό με την δική μας μάθηση. Εάν κάποιος μαθαίνει να παίζει ένα μουσικό όργανο με την επίβλεψη ενός δασκάλου, θα μάθει γρήγορα επαναχρησιμοποιώντας την εποπτευόμενη γνώση σημειώσεων, συγχορδιών και ρυθμών. Αν όμως ξεκινήσει να διδάσκει μόνος του τον εαυτό του, θα είναι πολύ πιο δύσκολο να αναγνωριστεί το σημείο εκκίνησης και η διαδικασία της εκμάθησης.

Με τη μη επιτήρηση σε ένα στυλ διδασκαλίας laissez-faire (αφήστε το/τα ελεύθερα), υπάρχει εκκίνηση από μια καθαρή βάση, με λιγότερη προκατάληψη και μπορεί να βρεθεί ακόμη και ένας νέος, καλύτερος τρόπος επίλυσης ενός προβλήματος. Ως εκ τούτου, αυτός είναι ο λόγος για τον οποίο η μη επιτηρούμενη μάθηση είναι επίσης γνωστή ως ανακάλυψη της γνώσης. Η μη εποπτευόμενη μάθηση είναι πολύ χρήσιμη κατά τη διεξαγωγή διερευνητικής ανάλυσης δεδομένων. Για να βρεθούν οι ενδιαφέρουσες δομές στα μη επισημασμένα δεδομένα, χρησιμοποιείται η εκτίμηση της πυκνότητας.

Η συνηθέστερη μορφή της είναι η ομαδοποίηση. Μεταξύ άλλων, υπάρχει επίσης μείωση των διαστάσεων, λανθάνοντα μεταβλητά μοντέλα και ανίχνευση ανωμαλιών. Οι πιο σύνθετες τεχνικές που δεν επιβλέπονται περιλαμβάνουν νευρωνικά δίκτυα όπως οι αυτόματοι κωδικοποιητές (Autencoders) που αναφέρονται παρακάτω και τα δίκτυα βαθιάς πεποίθησης (Deep Belief Networks).

### 2.3.1 Ομαδοποίηση (Clustering)

Η μη εποπτευόμενη μάθηση χρησιμοποιείται κυρίως για την ομαδοποίηση. Η ομαδοποίηση είναι η πράξη δημιουργίας ομάδων με διαφορετικά χαρακτηριστικά. Η ομαδοποίηση προσπαθεί να βρει διάφορες υποομάδες μέσα σε ένα σύνολο δεδομένων. Δεδομένου ότι πρόκειται για μάθηση χωρίς επίβλεψη, δεν υπάρχει περιορισμός σε κανένα σύνολο ετικετών και εκ τούτου ελευθερία επιλογής στο πόσα συμπλέγματα μπορούν να δημιουργηθούν. Η επιλογή ενός μοντέλου που έχει τον σωστό αριθμό ομάδων (πολυπλοκότητα) πρέπει να διεξαχθεί μέσω μιας εμπειρικής διαδικασίας επιλογής μοντέλου.



Εικόνα 2-2. Αποτελέσματα Ομαδοποίησης με χρήση K-means Clustering.

### 2.3.2 Μείωση διαστάσεων

Η μείωση των διαστάσεων αποσκοπεί στο να βρει τα πιο σημαντικά χαρακτηριστικά για να μειώσει τα αρχικά χαρακτηριστικά, με αποτέλεσμα να ορίζεται ένα μικρότερο, πιο αποτελεσματικό σύνολο, που όμως εξακολουθεί να κωδικοποιεί τα σημαντικά δεδομένα. Για παράδειγμα, στην πρόβλεψη του αριθμού επισκεπτών στην παραλία μπορούν να χρησιμοποιηθούν ως εισροές η θερμοκρασία, η ημέρα της εβδομάδας, ο μήνας και ο αριθμός των συμβάντων που προγραμματίζονται για εκείνη την ημέρα. Αλλά ο μήνας μπορεί να μην είναι πραγματικά σημαντικός για την πρόβλεψη του αριθμού των επισκεπτών.

Ανεξάρτητα χαρακτηριστικά όπως αυτό, μπορεί να προκαλέσουν σύγχυση σε αλγόριθμους μηχανικής μάθησης και να τους κάνουν λιγότερο αποτελεσματικούς και ακριβείς. Χρησιμοποιώντας τη μείωση των διαστάσεων, αναγνωρίζονται και χρησιμοποιούνται μόνο τα πιο σημαντικά χαρακτηριστικά. Η Ανάλυση Κύριων Συνιστωσών (Principal Component Analysis - PCA), είναι μια κοινώς χρησιμοποιούμενη τεχνική. Η Ανάλυση Κύριων Συνιστωσών (Principal Component Analysis - PCA), αποτελεί μία γραμμική μέθοδο συμπίεσης Δεδομένων, η οποία συνίσταται από τον επαναπροσδιορισμό των συντεταγμένων ενός συνόλου δεδομένων σε ένα άλλο σύστημα συντεταγμένων το οποίο θα είναι καταλληλότερο στην επικείμενη ανάλυση δεδομένων. Στην αρχική της μορφή, η Ανάλυση Κύριων Συνιστωσών (Principal Component Analysis - PCA), επιδιώκει έναν γραμμικό μετασχηματισμό για αυτό, ωστόσο, μπορεί να διαμορφωθεί για να επιτρέψει την αναζήτηση μη γραμμικών μετασχηματισμών.

### 2.3.3 Γενετικά μοντέλα

Τα γενετικά μοντέλα αποτελούν μια κατηγορία μη εποπτευόμενων μαθησιακών μοντέλων στα οποία παρέχονται δεδομένα κατάρτισης και παράγονται νέα δείγματα από την ίδια κατανομή.

Αυτά τα μοντέλα πρέπει να ανακαλύψουν και να μάθουν αποτελεσματικά την ουσία των δεδομένων για να προσπαθήσουν να δημιουργήσουν παρόμοια δεδομένα. Το μακροπρόθεσμο όφελος αυτού του τύπου μοντέλου είναι η ικανότητά του να μαθαίνει αυτόματα τα χαρακτηριστικά των δεδομένων.

Ένα κοινό παράδειγμα εφαρμογής γενετικών μοντέλων είναι σε ένα σύνολο δεδομένων εικόνων. Δεδομένου ενός συνόλου εικόνων, ένα γενετικό μοντέλο θα μπορούσε να δημιουργήσει ένα νέο σύνολο εικόνων με παρόμοια χαρακτηριστικά με το δεδομένο σύνολο.

### 2.3.4 Μη εποπτευόμενη βαθιά μάθηση

Δεν είναι έκπληξη το γεγονός ότι η μάθηση χωρίς επίβλεψη επεκτάθηκε επίσης στα νευρωνικά δίκτυα και στη βαθιά εκμάθηση. Αυτή η περιοχή είναι ακόμα νεοσύστατη, αλλά μια δημοφιλής εφαρμογή της βαθιάς μάθησης με ένα μη επιτηρούμενο τρόπο ονομάζεται Autoencoder (αυτόματοι κωδικοποιητές).

Οι αυτόματοι κωδικοποιητές ακολουθούν την ίδια φιλοσοφία με τους αλγόριθμους συμπίεσης δεδομένων που περιγράφηκαν παραπάνω, χρησιμοποιώντας ένα μικρότερο υποσύνολο χαρακτηριστικών που αντιπροσωπεύουν τα αρχικά μας δεδομένα.

Όπως ένα νευρωνικό δίκτυο, ένας αυτόματος κωδικοποιητής χρησιμοποιεί βάρη για να δοκιμάσει και να διαμορφώσει τις τιμές εισόδου σε μια επιθυμητή έξοδο. Όπως αναφέρθηκε παραπάνω, στην αρχική τής μορφή, η Ανάλυση Κύριων Συνιστωσών (Principal Component Analysis - PCA), επιδιώκει έναν γραμμικό μετασχηματισμό, ωστόσο, μπορεί να διαμορφωθεί για να επιτρέψει την αναζήτηση μη γραμμικών μετασχηματισμών. Ο αυτόματος κωδικοποιητής (Autoencoder) γενικεύει αυτήν την ιδέα, αναζητώντας έναν μετασχηματισμό (όχι απαραίτητα γραμμικό) από τον αρχικό μετασχηματισμό σε έναν χώρο χαμηλότερης διάστασης. Ο στόχος ενός αυτόματου κωδικοποιητή είναι να μάθει μια αναπαράσταση (κωδικοποίηση) για ένα σύνολο δεδομένων, συνήθως για μείωση διαστάσεων και από τη μειωμένη κωδικοποίηση μια αναπαράσταση όσο το δυνατόν πιο κοντά στην αρχική του είσοδο, εξ ου και το όνομά του.

### 2.3.5 Παραδείγματα Μη Επιβλεπόμενης Μάθησης

Στον πραγματικό κόσμο, η ομαδοποίηση έχει χρησιμοποιηθεί με επιτυχία στην αστρονομία για να ανακαλύψει ένα νέο είδος αστεριού διερευνώντας τις υποομάδες αστερών.

Στο μάρκετινγκ, χρησιμοποιείται συστηματικά για να συγκεντρώσει τους διάφορους πελάτες μίας εταιρείας σε παρόμοιες ομάδες με βάση τις συμπεριφορές και τα χαρακτηριστικά τους. Η εκμάθηση μέσω σύνδεσης χρησιμοποιείται για την εύρεση και συγκρότηση σχετικών στοιχείων.

Ένα κοινό παράδειγμα είναι η ανάλυση του καλαθιού αγοράς. Στην ανάλυση του καλαθιού αγοράς, οι κανόνες σύνδεσης επιτρέπουν την πρόβλεψη άλλων προϊόντων που ένας πελάτης είναι πιθανό να αγοράσει με

βάση αυτά που έχει ήδη τοποθετήσει στο καλάθι του. Οι περισσότερες δημοφιλείς ιστοσελίδες αγορών χρησιμοποιούν ακριβώς αυτή την τεχνική. Εάν για παράδειγμα τοποθετηθεί ένας φορητός υπολογιστής στο καλάθι αγορών μιας παραγγελίας, η ιστοσελίδα συνιστά και την αγορά και άλλων αντικειμένων όπως μια θήκη για φορητούς υπολογιστές και αυτό επιτυγχάνεται μέσω των κανόνων σύνδεσης τους.

## 2.4 Επιβλεπόμενη Μάθηση (Supervised Learning)

Παρακάτω θα δοθούν κάποια επιπλέον στοιχεία για το πως λειτουργεί η μέθοδος μηχανικής μάθησης στην οποία βασίζεται και η παρούσα εργασία. Στην εποπτευόμενη μάθηση, ο στόχος είναι να βρεθεί η χαρτογράφηση (τα πρότυπα) μεταξύ ενός συνόλου εισόδων και εξόδων. Για παράδειγμα, οι εισροές θα μπορούσαν να είναι η πρόγνωση του καιρού και οι εκροές, οι επισκέπτες της παραλίας. Ο στόχος της εποπτευόμενης μάθησης θα ήταν να μάθει τη χαρτογράφηση που περιγράφει τη σχέση μεταξύ θερμοκρασίας και αριθμού επισκεπτών στην παραλία.

Παραδείγματα δεδομένων με ετικέτα (label) παρέχονται από προηγούμενα ζεύγη εισόδου και εξόδου κατά τη διάρκεια της μαθησιακής διαδικασίας για να διδάξουν το μοντέλο πώς πρέπει να συμπεριφέρεται, επομένως, την «επίβλεψη» της μάθησης. Για το παράδειγμα της παραλίας, οι νέες εισροές μπορούν να τροφοδοτηθούν στον αλγόριθμο εκμάθησης ώστε να δώσει στη συνέχεια μια μελλοντική πρόβλεψη για τον αριθμό των επισκεπτών.

Η ικανότητά της να προσαρμόζεται σε νέες εισροές και να κάνει προβλέψεις είναι το κρίσιμο τμήμα γενίκευσης της μηχανικής μάθησης. Κατά την προσαρμογή, στόχος είναι να μεγιστοποιηθεί η γενίκευση, έτσι ώστε το υπό εποπτεία μοντέλο να ορίζει την πραγματική «γενική» υποκείμενη σχέση. Εάν το μοντέλο είναι υπερβολικά εκπαιδευμένο, προκαλούμε υπερβολική προσαρμογή (overfit) και το μοντέλο δεν είναι σε θέση να προσαρμοστεί σε νέες, προηγούμενως άγνωστες εισόδους.

Μια παρενέργεια που πρέπει να γνωρίζουμε στην εποπτευόμενη μάθηση είναι ότι η εποπτεία που παρέχουμε εισάγει προκατάληψη στη μάθηση. Το μοντέλο μπορεί μόνο να μιμείται ακριβώς αυτό που παρουσιάστηκε, γι' αυτό είναι πολύ σημαντικό να δείξουμε αξιόπιστα, αμερόληπτα παραδείγματα. Επίσης, η εποπτευόμενη μάθηση συνήθως απαιτεί πολλά δεδομένα πριν μάθει. Η απόκτηση αρκετών αξιόπιστων δεδομένων είναι συχνά το δυσκολότερο και ακριβότερο μέρος της χρήσης εποπτευόμενης μάθησης. Ως εκ τούτου, τα δεδομένα έχουν ονομαστεί «το νέο πετρέλαιο».

Η έξοδος από ένα εποπτευόμενο μοντέλο Machine Learning θα μπορούσε να είναι μια κατηγορία από ένα πεπερασμένο σύνολο π.χ. [χαμηλό, μεσαίο, υψηλό] για τον αριθμό των επισκεπτών στην παραλία:

Input [Θερμοκρασία=20] -> Model -> Output = [επισκέπτες=υψηλό]



Όταν συμβαίνει αυτό, το εποπτευόμενο μοντέλο αποφασίζει πώς να ταξινομήσει την είσοδο, κάτι που είναι γνωστό ως Ταξινόμηση (Classification) που αναλύεται στην ενότητα 2.4.1.

Εναλλακτικά, η έξοδος θα μπορούσε να είναι μια πραγματική συνεχής μεταβλητή (εξαγωγή ενός αριθμού που ανήκει σε ένα συνεχές διάστημα αριθμών):

$$\text{Input [Θερμοκρασία=20]} \rightarrow \text{Model} \rightarrow \text{Output = [επισκέπτες=300]}$$

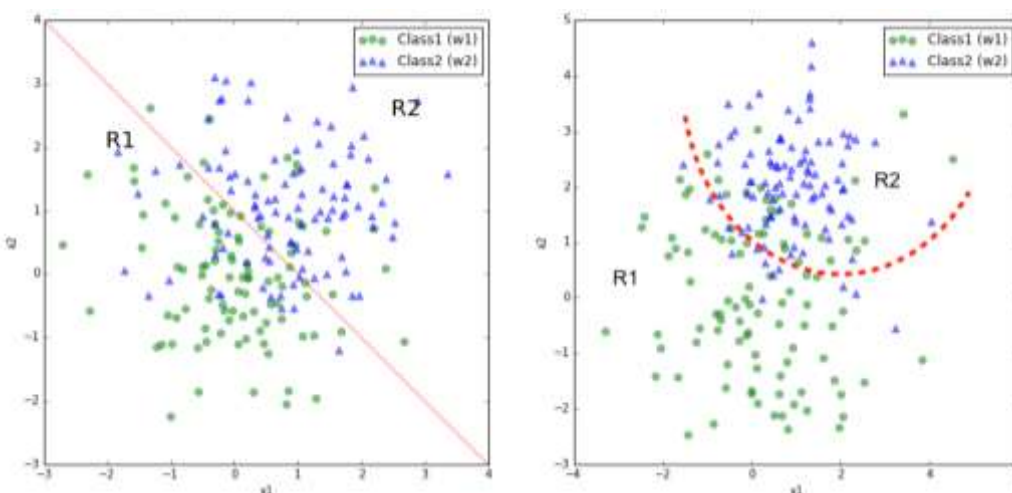
Αυτή η περίπτωση είναι γνωστή και ως Παλινδρόμηση (Regression) που αναλύεται στην ενότητα 2.4.2.

### 2.4.1 Ταξινόμηση (Classification)

Η ταξινόμηση χρησιμοποιείται για την ομαδοποίηση των παρόμοιων σημείων δεδομένων σε διαφορετικά τμήματα προκειμένου να ταξινομηθούν. Η Μηχανική Μάθηση χρησιμοποιείται για να βρεθούν οι κανόνες που εξηγούν τον τρόπο διαχωρισμού των διαφόρων σημείων δεδομένων. Αλλά ποιοι είναι οι μηχανισμοί δημιουργίας των κανόνων; Υπάρχουν πολλοί τρόποι δημιουργίας των κανόνων που όλοι εστιάζουν στη χρήση δεδομένων και απαντήσεων για να ανακαλύψουν κανόνες που διαχωρίζουν γραμμικά τα σημεία δεδομένων.

Η γραμμική διαχωρισσιμότητα είναι μια βασική ιδέα στη μηχανική μάθηση. Όλα αυτά τα μέσα γραμμικού διαχωρισμού προσπαθούν να απαντήσουν στο ερώτημα: «μπορούν τα διαφορετικά σημεία δεδομένων να διαχωριστούν από μια γραμμή». Έτσι, απλά, οι προσεγγίσεις ταξινόμησης προσπαθούν να βρουν τον καλύτερο τρόπο για να διαχωρίσουν τα σημεία δεδομένων με μια γραμμή.

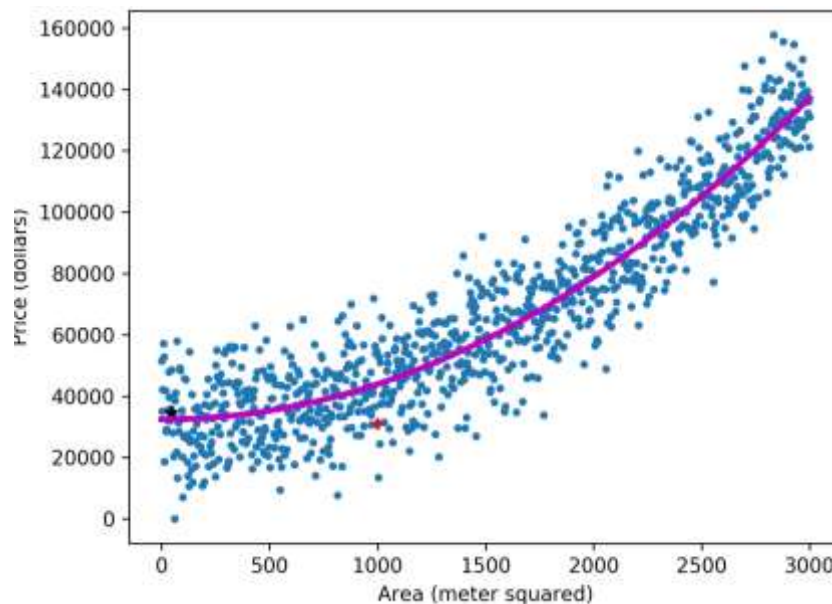
Οι γραμμές μεταξύ των κλάσεων είναι γνωστές ως όρια απόφασης. Ολόκληρη η περιοχή που επιλέγεται για τον ορισμό μιας κλάσης είναι γνωστή ως η επιφάνεια απόφασης. Η επιφάνεια απόφασης καθορίζει ότι εάν ένα σημείο δεδομένων εμπίπτει εντός των ορίων του, θα του δοθεί μια ορισμένη κλάση.



Εικόνα 2-3. Παράδειγμα διαχωριστικής γραμμής σε πρόβλημα Ταξινόμησης

#### 2.4.2 Παλινδρόμηση (Regression)

Η παλινδρόμηση είναι μια άλλη μορφή εποπτευόμενης μάθησης. Η διαφορά μεταξύ της ταξινόμησης και της παλινδρόμησης είναι ότι η παλινδρόμηση εξάγει έναν αριθμό παρά μια κατηγορία. Επομένως, η παλινδρόμηση είναι χρήσιμη όταν προβλέπουμε προβλήματα με βάση αριθμούς, όπως οι χρηματιστηριακές τιμές, η θερμοκρασία για μια δεδομένη ημέρα ή η πιθανότητα ενός γεγονότος.



Εικόνα 2-4. Παράδειγμα αλγορίθμου παλινδρόμησης (αξία σπιτιού συναρτήσει του εμβαδού)

#### 2.4.3 Πως λειτουργεί ένας αλγόριθμος εποπτευόμενης μάθησης

Στην εποπτευόμενη μάθηση η μάθηση προέρχεται από γνωστά δεδομένα (η ετικέτα που γνωρίζουμε για ένα σύνολο δεδομένων) για τη δημιουργία ενός μοντέλου από την πρόβλεψη της τάξης-στόχου ως εξόδου για τα δεδομένα εισόδου. Η εποπτευόμενη μάθηση είναι επίσης γνωστή ως εργασία εξόρυξης δεδομένων (data mining) και χρησιμοποιείται για την εξαγωγή μιας συνάρτησης από δεδομένα που έχουν επισημανθεί.

#### 2.4.4 Αλγόριθμοι επιβλεπόμενης μάθησης

Το πρώτο βήμα κάθε αλγόριθμου εποπτευόμενης μάθησης είναι η ανάλυση των δεδομένων εκπαίδευσης. Στο δεύτερο βήμα συνάγεται η συνάρτηση που μπορεί να χρησιμοποιηθεί για την απεικόνιση νέων παραδειγμάτων. Παρέχει εξόδους συνήθως σε μία από τις δύο ακόλουθες μορφές.

1. Οι εξοδοί παλινδρόμησης είναι πραγματικοί αριθμοί που υπάρχουν σε συνεχή χώρο.
2. Οι εξοδοί ταξινόμησης κατατάσσονται σε διακριτές κατηγορίες.



Όπως φαίνεται παραπάνω, τα προβλήματα κατά την ταξινόμηση (δυαδική ή πολυ-τάξη) και η παλινδρόμηση κατατάσσονται στην εποπτευόμενη μάθηση. Μερικοί από τους αλγόριθμους που λύνουν τέτοια προβλήματα αναφέρονται παρακάτω.

- Linear Regression
- Logistic Regression
- Polynomial Regression
- SVM for Regression
- Decision Trees
- Random Forest
- Support Vector Machine (SVM)
- Naive Bayes
- k-Nearest Neighbors

#### 2.4.5 Παραδείγματα Εποπτευόμενης Μάθησης

Η παλινδρόμηση χρησιμοποιείται στη χρηματοοικονομική διαπραγμάτευση για να βρει τα πρότυπα στα αποθέματα και άλλα περιουσιακά στοιχεία ώστε να μπορεί να αποφασίσει κάποιος πότε να αγοράσει ή να πουλήσει και να πραγματοποιήσει κέρδος. Η ταξινόμηση, χρησιμοποιείται ήδη για να ταξινομήσει αν ένα email που λαμβάνεται είναι θεμιτό ή spam (η μαζική αποστολή ηλεκτρονικών μηνυμάτων, εξωτερικών συνδέσμων ή άλλων, σε μια προσπάθεια προώθησης προϊόντων ή ιδεών που γίνεται σε μεγάλο αριθμό αποδεκτών).

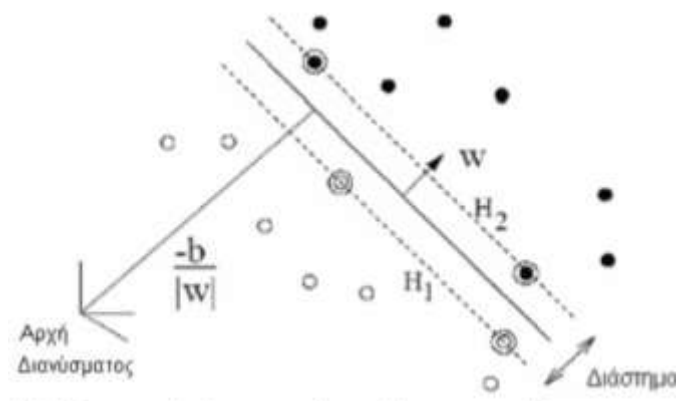
Τόσο η τεχνική μάθησης κατά την ταξινόμηση όσο και η παλινδρόμηση μπορούν να αναλάβουν και πιο πολύπλοκα καθήκοντα, για παράδειγμα, εργασίες που περιλαμβάνουν ομιλία και ήχο. Η ταξινόμηση εικόνων, η ανίχνευση αντικειμένων και τα ρομπότ συζήτησης είναι μερικά παραδείγματα.

Η ανάλυση των μηνυμάτων που προέρχονται από το σκοτεινό δίκτυο (Dark Web) συνιστά επίσης ένα πρόβλημα εποπτευόμενης μάθησης. Για την επεξεργασία του θα χρησιμοποιηθούν οι αλγόριθμοι SVM (Support Vector Machines), Gaussian Naïve Bayes, Decision Tree και Random Forest.

#### 2.4.6 Αλγόριθμος Support Vector Machines

Οι Support Vector Machines (SVM) - Μηχανές Διανυσμάτων Υποστήριξης (ΜΔΥ) συνιστούν μια τεχνική που χρησιμοποιείται για κατηγοριοποίηση (clustering) δεδομένων και εφαρμόζεται με πολύ μεγάλη επιτυχία στην κατηγοριοποίηση των αρχείων κειμένου. Για τον λόγο αυτό επιλέχθηκε για να επιλύσει το

πρόβλημα της εν λόγω εργασίας. Ο αλγόριθμος SVM βασίζεται στη θεωρία της στατιστικής μάθησης και στη διάσταση Vapnik-Chervonenkis (VC), που εισήγαγαν οι Vladimir Vapnik και Alexey Chervonenkis και επεκτάθηκε από τους Corinna Cortes και Vladimir Vapnik. Ο αλγόριθμος Support Vector Machine (SVM) συγκαταλέγεται ανάμεσα στους πιο αποδοτικούς κατηγοριοποιητές καθώς έχει μια μοναδική ικανότητα να χειρίζεται μεγάλα σύνολα χαρακτηριστών όπως για παράδειγμα μεγάλα σε όγκο είδη κειμένου. Ο SVM αλγόριθμος λειτουργεί ως εξής: χαρτογραφεί το εκπαιδευτικό σύνολο (training set) - στη περίπτωση μας εξετάζεται ένα σύνολο από μηνύματα τρομοκρατών και μη - σε ένα πιθανό πολυδιάστατο χώρο διανυσμάτων και προσπαθεί να εντοπίσει σε αυτό το χώρο ένα πεδίο το οποίο να διαχωρίζει τα θετικά από τα αρνητικά παραδείγματα. Έχοντας βρει ένα τέτοιο πεδίο, ο αλγόριθμος μπορεί να προβλέψει την κατηγοριοποίηση ενός αχαρακτήριστου παραδείγματος χαρτογραφώντας το στον χώρο που περιέχει τα χαρακτηριστικά και ψάχνοντας σε ποια πλευρά του διαχωριστικού πεδίου βρίσκεται. Πώς όμως επιλέγεται το διαχωριστικό πεδίο τη στιγμή που υπάρχουν πολλά υποψήφια; Το διαχωριστικό πεδίο που επιλέγεται συνιστά αυτό το οποίο διατηρεί το μεγαλύτερο διάστημα μεταξύ οποιουδήποτε σημείου στο συνολικό εκπαιδευόμενο σύνολο.



Εικόνα 2-5. Γραμμικώς διαχωρισμένα πεδία για την ευδιάκριτη γραμμική περίπτωση

Πιο αναλυτικά όλα τα διανύσματα εισόδου μπορούν να χωριστούν από τα πεδία  $H_1$  και  $H_2$ . Κάποια διανύσματα της περιοχής του χώρου της μίας κατηγορίας είναι πιο κοντά στην περιοχή του χώρου μιας άλλης κατηγορίας. Τα διανύσματα αυτά που βρίσκονται στο πεδίο  $H_1$  και στο πεδίο  $H_2$  ονομάζονται support vectors (διανύσματα υποστήριξης) και είναι κυκλωμένα στο παραπάνω σχήμα.

Ο στόχος του αλγόριθμου είναι να επιλέξει ένα διαχωριστικό πεδίο  $(w \cdot x_i + b) = 0$  το οποίο μεγιστοποιεί το διάστημα μεταξύ του  $H_1$  ( $w \cdot x_i + b = -1$ ) και του  $H_2$  ( $w \cdot x_i + b = 1$ ).

Αυτό υλοποιείται ως εξής : υποθέτουμε ότι όλα τα εκπαιδευτικά δεδομένα ικανοποιούν τους παρακάτω περιορισμούς:

$$y_i (w \cdot x_i + b) \geq 1$$

όπου  $y_i$  είναι η αντίστοιχη ζητούμενη τιμή. Αν  $y_i = 1$  τότε αυτό σημαίνει ότι το  $x_i$  ανήκει στην κατηγορία 1 και αν  $y_i = -1$  τότε το ανήκει στην κατηγορία 2.

Για ένα πεδίο  $(w \cdot x_i + b) = 0$ , η απόσταση από το πεδίο στην αρχή του διανύσματος  $x_i$  είναι  $|b|/\|w\|$ . Επομένως η απόσταση του  $H_1$  από την αρχή του διανύσματος είναι  $|b| + 1/\|w\|$  και αντίστοιχα η απόσταση του  $H_2$  από την αρχή του διανύσματος είναι  $|b| - 1/\|w\|$ . Άρα το διάστημα μεταξύ του  $H_1$  και του  $H_2$  είναι  $2/\|w\|$  οπότε και μπορεί να βρεθεί ένα ζεύγος πεδίων που θα δίνει το μέγιστο διάστημα ελαχιστοποιώντας την ποσότητα  $\|w\|^2$  λαμβάνοντας υπ' όψη τους περιορισμούς που αναφέρθηκαν προηγουμένως. Το πρόβλημα αυτό αποκαλείται «πρωτεύον πρόβλημα».

Στη συνέχεια ακολουθεί η μετάβαση σε ένα Lagrange-ιανό σχηματισμό του προβλήματος. Δοθέντων των θετικών πολλαπλασιαστών Lagrange  $a_i$  για κάθε περιορισμό ανισότητας ο Lagrange-ιανός αυτός σχηματισμός υλοποιείται ως εξής:

$$L_P = \frac{1}{2} \|w\|^2 - \sum_i a_i [y_i(w x_i - b) - 1]$$

Κατόπιν πρέπει να ελαχιστοποιηθεί η ποσότητα  $L_P$  ως προς τα  $w$  και  $b$ , και συγχρόνως να απαιτηθεί οι παράγωγοι του  $L_P$  ως προς όλα τα  $a_i$  να εξαφανιστούν. Αυτό είναι ισοδύναμο με το να λυθεί το παρακάτω δυαδικό πρόβλημα.

Απαιτώντας το διάνυσμα βαθμίδας (gradient) του  $L_P$  ως προς τα  $w$  και  $b$  να εξαφανιστεί δίνονται οι συνθήκες:

$$w(a) = \sum_i a_i y_i x_i$$

$$\sum_i a_i y_i = 0$$

Μπορούν να αντικατασταθούν στην εξίσωση

$$L_P = \frac{1}{2} \|w\|^2 - \sum_i a_i [y_i(w x_i - b) - 1]$$

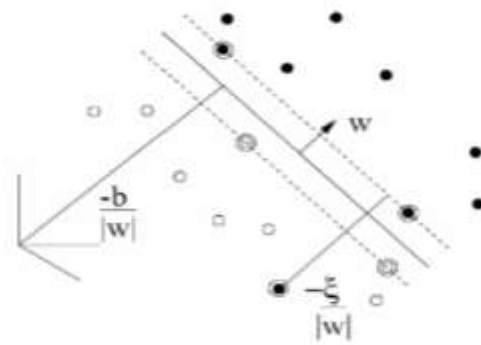
και να παραχθεί η εξής σχέση:

$$L_D = \sum_i a_i - \frac{1}{2} w(a) \cdot w(a)$$

Η εκπαίδευση με αυτόν τον αλγόριθμο (για την ευδιάκριτη, γραμμική περίπτωση) ωστόσο ισοδυναμεί με τη μεγιστοποίηση της ποσότητας  $L_D$  ως προς τα  $a_i$  λαμβάνοντας υπόψη τους περιορισμούς

$$\sum_i a_i y_i = 0$$

και  $\alpha_i \geq 0$ .



Εικόνα 2-6. Γραμμικώς διαχωρισμένα πεδία για την μη ευδιάκριτη γραμμική περίπτωση

Για τη μη ευδιάκριτη περίπτωση που φαίνεται στο παραπάνω σχήμα θα χαλαρώσουμε λίγο τους περιορισμούς  $y_i (w \cdot x_i + b) \geq 1$  εισάγοντας θετικές μεταβλητές  $\xi_i$ ,  $i = 1, 2, \dots, I$  στους περιορισμούς οι οποίοι γίνονται

$$y_i (w \cdot x_i + b) \geq 1 - \xi_i, \xi_i \geq 0, \forall i$$

Ωστόσο για να εμφανιστεί ένα σφάλμα στη μη ευδιάκριτη περίπτωση το αντίστοιχο  $\xi_i$  πρέπει να είναι μεγαλύτερο του ενός, οπότε το  $\sum_i \xi_i$  είναι το άνω όριο του αριθμού των εκπαιδευτικών λαθών. Ένας φυσικός τρόπος να αναθέσουμε ένα επιπλέον κόστος για σφάλματα είναι να αλλάξουμε την αντικειμενική συνάρτηση έτσι ώστε να ελαχιστοποιείται από  $\frac{\|w\|^2}{2}$  σε  $\frac{\|w\|^2}{2} + C \sum_i \xi_i$  όπου  $C$  είναι μια παράμετρος η οποία επιλέγεται από τον χρήστη. Ένα μεγάλο  $C$  αντιστοιχεί στην ανάθεση μεγαλύτερης ποινής στα σφάλματα. Όμοια με την ευδιάκριτη περίπτωση είναι επίσης ένα δευτερεύον προγραμματιστικό πρόβλημα και μπορεί να λυθεί μεγιστοποιώντας τη δυαδική μορφή:

$$L_D = \sum_i a_i - \frac{1}{2} w(a) \cdot w(a)$$

με περιορισμούς :

$$0 \leq \alpha_i \leq C$$

και

$$\sum_i a_i y_i = 0$$

δεδομένης της σχέσης

$$w(a) = \sum_i^{N_s} a_i y_i x_i$$

όπου  $N_s$  είναι ο αριθμός των support vectors. Ωστόσο η μόνη διαφορά από τη βέλτιστη περίπτωση πεδίων είναι το ότι το  $\alpha_i$  έχει πλέον ένα άνω όριο το  $C$ . Οι μέθοδοι μπορούν να γενικευτούν στην περίπτωση

που η συνάρτηση απόφασης (decision function) είναι μια μη γραμμική συνάρτηση των δεδομένων. Ας υποθέσουμε ότι πρώτα χαρτογραφούμε τα δεδομένα σε κάποιο άλλο πολυδιάστατο χώρο  $H$ , χρησιμοποιώντας ένα μη γραμμικό μετασχηματισμό  $Z_i = \Phi(X_i)$ . Τότε το ίδιο πρόβλημα μπορεί να μορφοποιηθεί σε ένα πολυδιάστατο χώρο. Τα εσωτερικά γινόμενα της μορφής  $\Phi(X_i) \cdot \Phi(X_j)$  θα χρησιμοποιούνται για την εκπαίδευση του SVM. Συνήθως η συνάρτηση  $\Phi$  θεωρείται άγνωστη ενώ αντιθέτως ορίζεται μία συνάρτηση πυρήνα  $k(x, \mathbf{x}) = \Phi(x) \cdot \Phi(\mathbf{x})$ . Πολλοί γνωστοί πυρήνες περιλαμβάνουν:

Πολυωνυμικές συναρτήσεις:  $K(X,Y) = (1+X \cdot Y)^d$

Συναρτήσεις Radial Basis(RBF)

Radial Basis Functions:  $K(X,Y) = \exp(-||X - Y||^2/2\sigma^2)$

Σιγμοειδείς:  $K(X,Y) = \tanh(k_1 X \cdot Y + k_2)$

#### 2.4.7 Bayes Αλγόριθμοι

Από τους σημαντικότερους στόχους της πιθανοθεωρητικής προσέγγισης στη μάθηση είναι η εύρεση της πιο πιθανής υπόθεσης του χώρου υποθέσεων  $H$ , δεδομένου ενός σώματος εκπαίδευσης  $D$  και της όποιας γνώσης ενδεχομένως διαθέτουμε για τις πιθανότητες των διαφόρων υποθέσεων  $h \in H$ . Η πιθανότητα ισχύος μιας υπόθεσης  $h$  δεδομένου ενός συνόλου στιγμιότυπων  $D$  δίδεται από νόμο του Bayes:

$$\Pr(h | D) = \frac{\Pr(h) \Pr(D | h)}{\Pr(D)}$$

όπου:

- $\Pr(h)$  : η εκ των προτέρων πιθανότητα ισχύος της  $h$ , χωρίς να προηγηθεί παρατήρηση των δεδομένων του  $D$ .
- $\Pr(D | h)$  : η δεσμευμένη πιθανότητα που εκφράζει το ενδεχόμενο παρατήρησης των δεδομένων του  $D$ , ισχυούσης της  $h$  (πιθανοφάνειας –likelihood).
- $\Pr(D)$  : η εκ των προτέρων πιθανότητα παρατήρησης των δεδομένων του  $D$ . Ο συγκεκριμένος όρος απλοποιείται και δε συμμετέχει στους υπολογισμούς.
- $\Pr(h | D)$  : η ζητούμενη εκ των υστέρων πιθανότητα ισχύος της  $h$  δεδομένης της παρατήρησης των δεδομένων του  $D$ .

Η αναζήτηση επομένως της πιο πιθανής υπόθεσης  $h$  δεδομένου του  $D$  ανάγεται στην εύρεση της υπόθεσης εκείνης με τη μεγαλύτερη εκ των υστέρων πιθανότητα (maximum a-posteriori ή MAP hypothesis). Ορίζουμε την υπόθεση αυτή ως:

$$h_{\text{map}} = \operatorname{argmax} \{ \Pr(h | D) \} = \operatorname{argmax} \left\{ \frac{\Pr(h) \Pr(D | h)}{\Pr(D)} \right\} = \operatorname{argmax} \{ \Pr(h) \Pr(D | h) \}$$

Ο παραπάνω αλγόριθμος συναντάται στη βιβλιογραφία με το όνομα Βέλτιστος Ταξινομητής Bayes, καθώς αποδεικνύεται θεωρητικά πως είναι σε θέση να υποδείξει το άνω φράγμα των επιδόσεων ενός συστήματος ταξινόμησης για ένα συγκεκριμένο πρόβλημα.

Για το πρόβλημα της ταξινόμησης, χρησιμοποιούμε διανυσματική αναπαράσταση των δεδομένων, έστω:

- $C$ : τυχαία μεταβλητή που δείχνει την κλάση ενός στιγμιότυπου.
- $X$ : διάνυσμα τυχαίων μεταβλητών που δείχνει τις τιμές των παρατηρούμενων χαρακτηριστικών.
- $c$ : μια συγκεκριμένη ετικέτα κλάσης.
- $x$ : ένα συγκεκριμένο παρατηρούμενο διάνυσμα

Έχοντας ένα στιγμιότυπο δοκιμής  $x$  για ταξινόμηση, χρησιμοποιούμε το νόμο του Bayes για να υπολογίσουμε την εκ των υστέρων πιθανότητα για κάθε κλάση δεδομένου του διανύσματος  $x$ , και επιλέγουμε την μεγαλύτερη από αυτές:

$$\begin{aligned} \operatorname{argmax} \{p(C=c \mid X=x)\} &= \operatorname{argmax} \left\{ \frac{p(C=c)p(X=x \mid C=c)}{p(X=x)} \right\} \\ &= \operatorname{argmax} \{p(C=c)p(X=x \mid C=c)\} \end{aligned}$$

Εδώ  $X = x$  αναπαριστά το γεγονός  $X_1 = x_1, X_2 = x_2, \dots, X_k = x_k$  οπότε:

$$\operatorname{argmax} \{p(C=c \mid X=x)\} = \operatorname{argmax} \{p(C=c)p(\bigcap X_i = x_i \mid C = c)\}$$

Όπου οι πιθανότητες στο δεξί μέλος της ισότητας εκτιμώνται από τα ταξινομημένα στιγμιότυπα του σώματος εκπαίδευσης.

#### 2.4.8 Naive Bayes

Ο Naive Bayes ταξινομητής προσφέρει μια απλή πιθανοθεωρητική προσέγγιση στα προβλήματα μάθησης με επίβλεψη, όπου στόχος μας είναι να προβλεφθεί επακριβώς η κατηγορία-κλάση των στιγμιότυπων δοκιμής χρησιμοποιώντας ταξινομημένα στιγμιότυπα εκπαίδευσης που περιλαμβάνουν την πληροφορία της κλάσης που ανήκουν.

Βασίζεται σε δυο σημαντικές ελκaiνευτικές υποθέσεις. Συγκεκριμένα, υποθέτει ότι κάθε χαρακτηριστικό των στιγμιότυπων είναι στοχαστικά ανεξάρτητο των υπολοίπων, δεδομένης της κλάσης και ότι δεν υπάρχουν άλλα κρυφά χαρακτηριστικά που να επηρεάζουν την διαδικασία της πρόβλεψης. Έτσι η πιθανότητα της σχέσης  $\operatorname{argmax} \{p(C=c \mid X=x)\} = \operatorname{argmax} \{p(C=c)p(\bigcap X_i = x_i \mid C = c)\}$  μετατρέπεται σε γινόμενο πιθανοτήτων. Οπότε:

$$\operatorname{argmax} \{p(C=c \mid X=x)\} = \operatorname{argmax} \{p(C=c)\prod p(X_i = x_i \mid C=c)\}$$

Ο παράγοντας  $p(C=c) =$  υπολογίζεται βάσει της συχνότητας εμφάνισης της κλάσης  $c$  στα στιγμιότυπα του σώματος εκπαίδευσης. Οι δεσμευμένες πιθανότητες  $p(X_i=x_i | C=c)$  υπολογίζονται ανάλογα με το αν το χαρακτηριστικό  $X_i$  είναι διακριτό ή συνεχές. Για τα διακριτά χαρακτηριστικά των διανυσμάτων, εκείνα δηλαδή που παίρνουν διακριτές τιμές, η πιθανότητα αυτή είναι ένας πραγματικός αριθμός μεταξύ 0 και 1 που αντιπροσωπεύει την πιθανότητα το εκάστοτε χαρακτηριστικό  $X_i$  να πάρει την τιμή  $x_i$  δεδομένης της κλάσης  $c$ . Για τα συνεχή χαρακτηριστικά, θεωρούμε ότι οι τιμές ακολουθούν μια πιθανοτική κατανομή (ξεχωριστή για κάθε χαρακτηριστικό), η οποία προσεγγίζεται από τα διανύσματα εκπαίδευσης. Η συνήθης θεώρηση είναι οι τιμές των χαρακτηριστικών είναι κανονικά κατανομημένες. Οπότε για συνεχή χαρακτηριστικά έχουμε:

$$p(X_i=x_i | C=c) = g(x_i, \mu_{i,c}, \sigma_{i,c})$$

Όπου  $g(x, \mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$  η συνάρτηση πυκνότητας πιθανότητας μια κανονικής (Gaussian) κατανομής.

Το παραπάνω μοντέλο μας αφήνει ένα μικρό αριθμό παραμέτρων που θα εκτιμηθούν από το σώμα εκπαίδευσης. Για κάθε κλάση και διακριτό χαρακτηριστικό, χρειάζεται να εκτιμηθεί η πιθανότητα το χαρακτηριστικό να πάρει κάθε τιμή από τις δυνατές διακριτές τιμές του, δεδομένης της κλάσης. Για κάθε κλάση και συνεχές χαρακτηριστικό, χρειάζεται να εκτιμηθεί η μέση τιμή και η τυπική απόκλιση της κατανομής που ακολουθούν οι τιμές του χαρακτηριστικού, δεδομένης της κλάσης.

#### 2.4.8.1 Παράδειγμα ενός Bayesian ταξινόμητή

Έστω ότι έχουμε στη διάθεσή μας στιγμιότυπα από τη βάση δεδομένων ενός καταστήματος με ηλεκτρονικά είδη (Εικόνα 2-6). Κάθε στιγμιότυπο αποτελείται από τα γνωρίσματα age, income, student, credit rating και ανήκει σε κάποια από τις δύο κλάσεις του προβλήματος (Yes, No).

RID	age	income	student	Credit_rating	Class: buys_computer
1	<=30	high	No	fair	No
2	<=30	high	No	excellent	No
3	31...40	high	No	fair	Yes
4	>40	medium	no	Fair	Yes
5	>40	low	yes	Fair	Yes
6	>40	low	yes	Excellent	No
7	31...40	low	yes	Excellent	Yes
8	<=30	medium	no	fair	No
9	<=30	low	yes	Fair	Yes
10	>40	medium	yes	fair	Yes
11	<=30	medium	yes	Excellent	Yes
12	31...40	medium	no	Excellent	Yes
13	31...40	high	yes	fair	Yes
14	>40	medium	no	Excellent	No

Εικόνα 2-7. Στιγμιότυπα από τη βάση δεδομένων ενός καταστήματος με ηλεκτρονικά είδη



Έστω ότι θέλουμε να ταξινομηθεί το ακόλουθο άγνωστο στιγμιότυπο του προβλήματος:

$X = (\text{age} = "<=30", \text{income} = \text{"medium"}, \text{student} = \text{"yes"}, \text{credit\_rating} = \text{"fair"})$

Αρχικά υπολογίζονται οι εκ των προτέρων (a priori) πιθανότητες των δύο κλάσεων του προβλήματος:

$$P(\text{buys\_computer} = \text{"yes"}) = 9/14 = 0.643$$

$$P(\text{buys\_computer} = \text{"no"}) = 5/14 = 0.357$$

Στη συνέχεια, υπολογίζονται οι υπό συνθήκη πιθανότητες για κάθε γνώρισμα για όλες τις κλάσεις του προβλήματος:

$$P(\text{age} = "<=30" | \text{buys\_computer} = \text{"yes"}) = 2/9 = 0.222$$

$$P(\text{age} = "<=30" | \text{buys\_computer} = \text{"no"}) = 3/5 = 0.600$$

$$P(\text{income} = \text{"medium"} | \text{buys\_computer} = \text{"yes"}) = 4/9 = 0.444$$

$$P(\text{income} = \text{"medium"} | \text{buys\_computer} = \text{"no"}) = 2/5 = 0.400$$

$$P(\text{student} = \text{"yes"} | \text{buys\_computer} = \text{"yes"}) = 6/9 = 0.667$$

$$P(\text{student} = \text{"yes"} | \text{buys\_computer} = \text{"no"}) = 1/5 = 0.200$$

$$P(\text{credit\_rating} = \text{"fair"} | \text{buys\_computer} = \text{"yes"}) = 6/9 = 0.667$$

$$P(\text{credit\_rating} = \text{"fair"} | \text{buys\_computer} = \text{"no"}) = 2/5 = 0.400$$

Χρησιμοποιώντας τις παραπάνω πιθανότητες υπολογίζουμε την πιθανότητα το άγνωστο στιγμιότυπο  $X$  να ανήκει σε κάποια από τις δύο κλάσεις του προβλήματος:

$$P(X | \text{buys\_computer} = \text{"yes"}) = 0.222 * 0.444 * 0.667 * 0.667 = 0.044$$

$$P(X | \text{buys\_computer} = \text{"no"}) = 0.600 * 0.400 * 0.200 * 0.400 = 0.019$$

$$P(X | \text{buys\_computer} = \text{"yes"}) * P(\text{buys\_computer} = \text{"yes"}) = 0.044 * 0.643 = 0.028$$

$$P(X | \text{buys\_computer} = \text{"no"}) * P(\text{buys\_computer} = \text{"no"}) = 0.019 * 0.357 = 0.007$$

Η κλάση με τη μεγαλύτερη πιθανότητα, εν προκειμένω η κλάση “Yes”, είναι η απάντηση που ψάχνουμε.

#### 2.4.9 Δένδρα απόφασης (Decision trees)

Τα δέντρα απόφασης είναι πολύ ισχυρά εργαλεία που χρησιμοποιούνται ευρέως για τις περιπτώσεις της ταξινόμησης και της πρόβλεψης. Ένα δέντρο απόφασης αντιπροσωπεύει μια σειρά από IF / THEN κανόνες ξεκινώντας από τη ρίζα του δέντρου και καταλήγοντας στα φύλλα του. Οι εσωτερικοί κόμβοι ενός δέντρου απόφασης περιέχουν τα γνωρίσματα του προβλήματος, οι ακμές περιέχουν τις δυνατές τιμές των γνωρισμάτων και τα φύλλα περιέχουν τις πιθανές κλάσεις του προβλήματος. Απαραίτητο για την κατασκευή ενός δέντρου

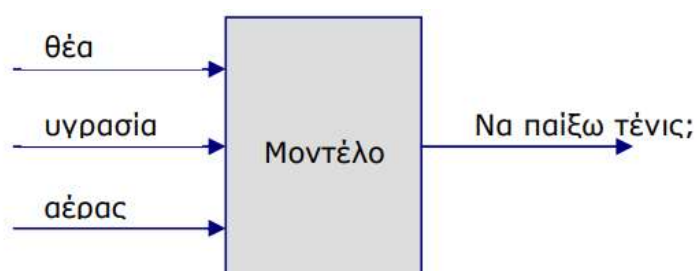


απόφασης είναι ένα σύνολο από στιγμιότυπα εκπαίδευσης, κάθε στιγμιότυπο του οποίου περιγράφεται από κάποια γνωρίσματα και την κλάση του προβλήματος στην οποία ανήκει.

Η διαδικασία που ακολουθούν οι αλγόριθμοι κατασκευής ενός δέντρου απόφασης συνοψίζεται στα ακόλουθα: Ξεκινώντας από τη ρίζα του δέντρου ο αλγόριθμος διασπά το σύνολο των στιγμιότυπων εκπαίδευσης σε υποσύνολα με βάση τη βέλτιστη ιδιότητα (best attribute) του κόμβου – η βέλτιστη ιδιότητα ενός κόμβου καθορίζεται από κάποιο κριτήριο όπως το information gain, το gain ratio κ.ο.κ. Έτσι προκύπτει ένα πλήθος υποσυνόλων που το καθένα περιέχει λιγότερα παραδείγματα από το αρχικό σύνολο. Για καθένα απ' αυτά τα επιμέρους υποσύνολα εφαρμόζεται επαναληπτικά η παραπάνω διαδικασία χρησιμοποιώντας τα εναπομείναντα γνωρίσματα, οπότε η διάσπαση των στιγμιότυπων προχωρά και σταματά όταν όλα τα στιγμιότυπα του υποσυνόλου ανήκουν στην ίδια κλάση ή έχουν εξαντληθεί όλα τα γνωρίσματα.

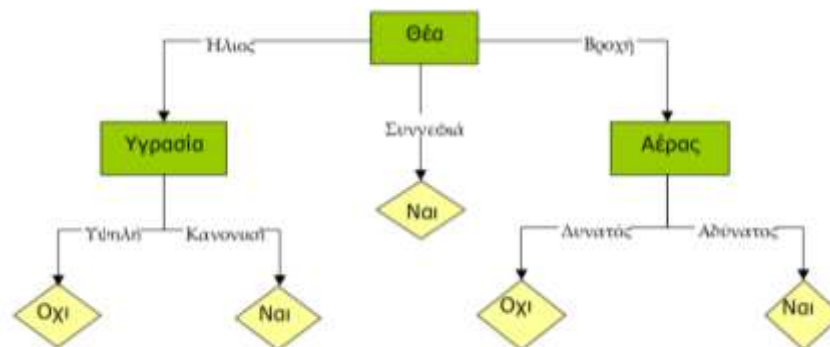
Στην ουσία πρόκειται για εφαρμογή της μεθόδου «Διαιρεί και βασίλευε». Εκτός από το σύνολο των στιγμιότυπων εκπαίδευσης υπάρχει και το σύνολο των στιγμιότυπων ελέγχου με βάση τα οποία ελέγχεται η απόδοση του δέντρου, δηλαδή η ακρίβεια με την οποία το κατασκευασμένο δέντρο απαντά στο πρόβλημα της ταξινόμησης. Στην περίπτωση αυτή δίνονται ως είσοδος στο δέντρο οι τιμές των γνωρισμάτων του στιγμιότυπου ελέγχου και αναμένεται ως απάντηση η τάξη του στιγμιότυπου. Το πλήθος των λανθασμένων απαντήσεων (δηλαδή τα στιγμιότυπα στα οποία το δέντρο απάντησε διαφορετική κλάση από την πραγματική) καθορίζει την ακρίβεια του δέντρου. Η διαδικασία που ακολουθείται προκειμένου να ταξινομηθεί ένα νέο στιγμιότυπο του προβλήματος είναι η ακόλουθη: διατρέχεται το δέντρο από τη ρίζα προς τα φύλλα ακολουθώντας τα κατάλληλα μονοπάτια. Κάθε φορά η επιλογή του μονοπατιού καθορίζεται εφαρμόζοντας τη συνθήκη ελέγχου κάθε κόμβου στις τιμές στιγμιότυπου. Όταν η διάτρεξη του δέντρου καταλήξει σε κάποιο φύλλο, η κλάση αυτού είναι και η ζητούμενη κλάση του στιγμιότυπου.

Έστω για παράδειγμα το κλασικό πρόβλημα που προσπαθεί να απαντήσει στο ερώτημα «Παίξε τένις» και το οποίο έχει δύο κλάσεις: «Ναι» και «Όχι» (Εικόνα 2-8). Η απάντηση στο πρόβλημα εξαρτάται από τους εξής παράγοντες: τη Θέα (με πιθανές τιμές: ήλιος, βροχή, συννεφιά), την Υγρασία (με πιθανές τιμές: υψηλή, κανονική) και τον Αέρα (με πιθανές τιμές: δυνατός, αδύνατος).



Εικόνα 2-8. Το πρόβλημα «Παίξε τένις».

Στο παρακάτω σχήμα (Εικόνα 2-9) φαίνεται το δέντρο απόφασης του προβλήματος. Περιέχει 3 εσωτερικούς κόμβους, σε κάθε κόμβο γίνεται έλεγχος ως προς κάποιο από τα γνωρίσματα του προβλήματος, ενώ στα φύλλα του περιέχονται οι κλάσεις του προβλήματος.



Εικόνα 2-9. Δέντρο απόφασης για το πρόβλημα «παίξε τένις».

Τα δέντρα απόφασης χρησιμοποιούνται ευρέως τόσο από την επιστημονική κοινότητα όσο και από τη βιομηχανία και αρκετοί αλγόριθμοι έχουν αναπτυχθεί για το σκοπό αυτό, όπως για παράδειγμα ο CART, ο ID3, ο C4.5, ο ITI κ.α.

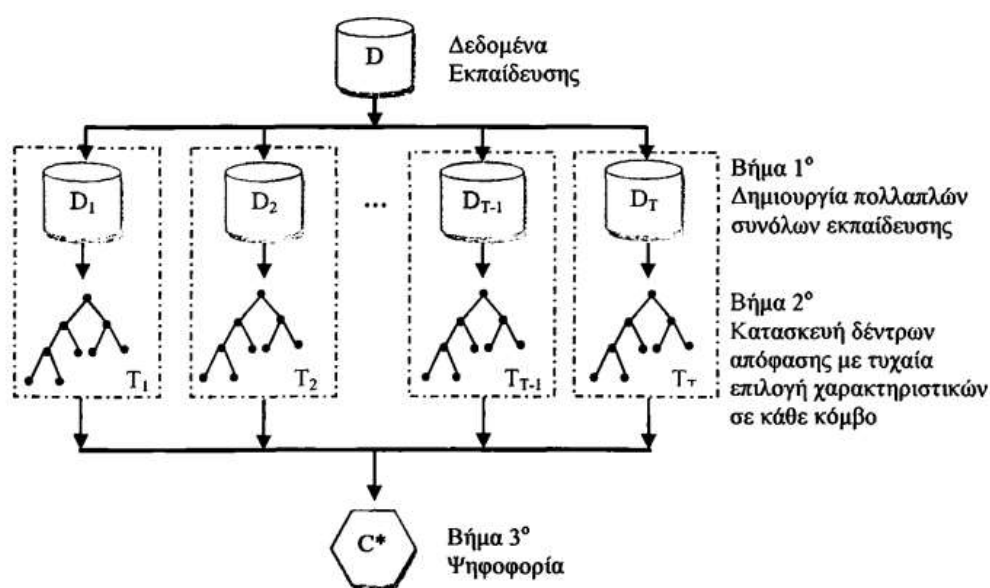
Ένας από τους βασικούς λόγους για τους οποίους τα δέντρα απόφασης είναι τόσο δημοφιλή είναι το γεγονός ότι πέραν της ικανότητάς τους να απαντούν με ικανοποιητική ακρίβεια σε προβλήματα ταξινόμησης και πρόβλεψης, μας βοηθούν και με την ευκολία με την οποία μπορούν να διατυπωθούν σε φυσική γλώσσα ή ακόμα και σε γλώσσες προσπέλασης δεδομένων, όπως η SQL (Structured Query Language), έτσι ώστε να είναι εύκολα κατανοητά από τους ανθρώπους.

#### 2.4.10 Τυχαία Δάση (Random Forests)

Τα τυχαία δάση (random forests) είναι ένας συνδυασμός δέντρων απόφασης έτσι ώστε κάθε δέντρο να εξαρτάται από τις τιμές ενός τυχαίου φορέα δειγματοληψίας ανεξάρτητα. Το σφάλμα γενίκευσης για τα δάση συγκλίνει σε ένα όριο καθώς ο αριθμός των δένδρων στο δάσος γίνεται μεγάλος. Το σφάλμα γενίκευσης ενός δάσους ταξινομητών δέντρων εξαρτάται από τη δύναμη των μεμονωμένων δέντρων στο δάσος και τη συσχέτιση μεταξύ τους. Οι γενικές εκτιμήσεις σφαλμάτων παρακολούθησης, δύναμη (strength) και συσχετισμός, χρησιμοποιούνται για να δείξουν την αύξηση του αριθμού των χαρακτηριστικών που χρησιμοποιούνται κατά τη διαδικασία διαχωρισμού (splitting). Χρησιμοποιούνται επίσης εσωτερικές εκτιμήσεις για την μέτρηση της αξίας μιας μεταβλητής. Αυτές οι πρακτικές είναι επίσης εφαρμοστέες στην μέθοδο της παλινδρόμησης.

Σημαντικές βελτιώσεις στην ακρίβεια ταξινόμησης έχουν προκύψει από την ανάπτυξη ενός συνόλου δέντρων τα οποία ψηφίζουν για τη δημοφιλέστερη τάξη μεταξύ τους. Για να αναπτυχθούν αυτά τα σύνολα, συχνά παράγονται τυχαία διανύσματα που διέπουν την ανάπτυξη κάθε δέντρου στο εκπαιδευτικό σύνολο. Ένα πρώιμο παράδειγμα είναι οι bagging predictors (Breiman [1996]), όπου για να αναπτυχθεί κάθε δέντρο από μια τυχαία επιλογή (χωρίς αντικατάσταση) χρησιμοποιούνται δείγματα από το σύνολο εκπαίδευσης (training set).

Τα τυχαία δάση αποτελούν μια ειδική κατηγορία των συνδυαστικών μεθόδων ταξινόμησης η οποία χρησιμοποιεί για ταξινομητές δέντρα απόφασης.



Εικόνα 2-10. Διαδικασία Κατασκευής των Τυχαίων Δασών.

Για την κατασκευή ενός δέντρου απόφασης ανατίθεται αρχικά στη ρίζα του το σύνολο των δειγμάτων εκπαίδευσης. Κάθε ενδιάμεσος κόμβος περιέχει υποσύνολο των δειγμάτων το οποίο μέσω της εφαρμογής ενός κατάλληλου ελέγχου διαχωρίζεται σε δυο ή περισσότερα μικρότερα υποσύνολα (παιδιά) στο επόμενο επίπεδο. Ο έλεγχος συνήθως αφορά ένα υποσύνολο των χαρακτηριστικών των δειγμάτων εκπαίδευσης. Η επιλογή του καλύτερου διαχωρισμού γίνεται σύμφωνα με ένα κατάλληλο μέτρο όπως π.χ. Gini index, εντροπία, misclassification error. Τα δέντρα του δάσους αναπτύσσονται στο μέγιστο μέγεθος τους, χωρίς κλάδεμα. Η μέθοδος Bagging χρησιμοποιώντας για ταξινομητές δέντρα απόφασης αποτελεί μια ειδική κατηγορία των Random Forests. Σε αυτή την περίπτωση η τυχαιότητα ενσωματώνεται στο μοντέλο μέσω της τυχαίας επιλογής  $N$  παραδειγμάτων εκπαίδευσης, με επανατοποθέτηση από το αρχικό σύνολο εκπαίδευσης. Η διαδικασία ταξινόμησης «άγνωστων» παραδειγμάτων πραγματοποιείται μέσω της διάσχισης των δέντρων του δάσους ξεκινώντας από τη ρίζα και καταλήγοντας σε ένα από τα φύλλα του δέντρου και στη συνέχεια

συνδυάζοντας τις προβλέψεις των ταξινομητών σύμφωνα με ένα πλειοψηφικό σύστημα ψηφοφορίας (majority voting scheme). Κάθε παράδειγμα ανατίθεται στην κατηγορία με τη μεγαλύτερη συχνότητα.

Θεωρητικά έχει αποδειχθεί ότι το σφάλμα γενίκευσης (generalization error) των τυχαίων δασών συγκλίνει στην ακόλουθη έκφραση όταν ο αριθμός των δέντρων είναι αρκετά μεγάλος.

$$\text{Generalization\_error} \leq \rho \frac{(1-s^2)}{s^2}$$

Όπου  $\rho$  είναι η μέση συσχέτιση μεταξύ των ταξινομητών και  $s$  μια ποσότητα η οποία υπολογίζει τη «δύναμη» των ταξινομητών. Η δύναμη (strength) ενός συνόλου ταξινομητών αναφέρεται στη μέση απόδοση των ταξινομητών  $M(X,Y)$ . Η απόδοση υπολογίζεται πιθανοτικά ως το περιθώριο του ταξινομητή  $M\{X, Y\} = P(\hat{Y}=Y) - \max P(\hat{Y}=Z)$ , όπου  $\hat{Y}$  είναι η προβλεπόμενη κατηγορία του  $X$  σύμφωνα με έναν ταξινομητή ο οποίος έχει κατασκευαστεί από κάποιο τυχαίο διάνυσμα  $\theta$ . Όσο πιο μεγάλο είναι το περιθώριο, τόσο πιο πιθανό είναι ο ταξινομητής να προβλέψει σωστά την κατηγορία του παραδείγματος  $X$ . Δηλαδή, όσο μεγαλώνει η συσχέτιση μεταξύ των δέντρων ή αντίστοιχα όσο η δύναμη του δάσους μειώνεται, το όριο του σφάλματος γενίκευσης τείνει να αυξάνεται. Η τυχαιότητα βοηθάει στη μείωση της συσχέτισης μεταξύ των δέντρων απόφασης και κατ' επέκταση στη βελτίωση του σφάλματος γενίκευσης του συνδυαστικού μοντέλου. Η παραπάνω σχέση εξηγεί το λόγο για τον οποίο τα Random Forests δεν εμφανίζουν φαινόμενα υπερεκπαίδευσης κατά την προσθήκη περισσότερων δέντρων αλλά αντιθέτως τείνουν να περιορίζουν το σφάλμα γενίκευσης.

Παρακάτω αναφέρονται συνοπτικά οι ιδιότητες και τα κύρια πλεονεκτήματα των Random Forests:

- Μπορούν να εκπαιδευτούν σε σύνολα δεδομένων υψηλής διάστασης όπως είναι τα κείμενα και οι εικόνες, χωρίς να εμφανίσουν σημαντικό βαθμό υπερεκπαίδευσης.
- Εξαιτίας του μεγάλου πλήθους δέντρων στο δάσος, το σφάλμα γενίκευσης είναι περιορισμένο. Αυτό έχει ως αποτέλεσμα τη μη εμφάνιση φαινομένων υπερεκπαίδευσης.
- Μη επαναληπτική διαδικασία εκπαίδευσης, ο αλγόριθμος ολοκληρώνεται σε σταθερό αριθμό βημάτων.
- Η τυχαία επιλογή ενός υποσυνόλου των χαρακτηριστικών για τη διαμέριση των παραδειγμάτων κάθε ενδιάμεσου κόμβου ελαττώνει τη συσχέτιση ανάμεσα στα δέντρα και διατηρεί την πόλωση (bias) σε χαμηλά επίπεδα καθώς τα δέντρα αναπτύσσονται χωρίς κλάδεμα. Χρησιμοποιώντας ένα σύνολο δέντρων απόφασης μειώνεται και η διακύμανση (variance).
- Η διάσχιση ενός δέντρου από ένα παράδειγμα ξεκινώντας από τη ρίζα και καταλήγοντας σε έναν από τους τερματικούς κόμβους είναι λογαριθμική συνάρτηση του πλήθους των φύλλων του.

- Παρουσιάζουν ανεικτικότητα ως προς το θόρυβο και τα αριθμητικά σφάλματα στα δεδομένα εκπαίδευσης (π.χ. απόκριση μέρους του αντικειμένου, ελλιπή δεδομένα).
- Για την διάσχιση κάθε δέντρου περίπου το 1/3 των παραδειγμάτων δεν επιλέγεται για εκπαίδευση. Αυτά τα παραδείγματα καλούνται Out-of-Bag παραδείγματα και μπορούν να χρησιμοποιηθούν για την εκτίμηση της πιθανότητας σφάλματος, εξαλείφοντας την ανάγκη ύπαρξης ενός συνόλου ελέγχου ή εφαρμογής της τεχνικής cross-validation.
- Παράγει μια εσωτερική αμερόληπτη εκτίμηση του σφάλματος γενίκευσης καθώς εξελίσσεται η διαδικασία κατασκευής του δέντρου.
- Υπάρχει η δυνατότητα παράλληλης επαγωγής των δέντρων, σε αντίθεση με την μέθοδο Boosting.
- Αναζητά το καλύτερο διαχωρισμό σε ένα μικρό υποσύνολο των χαρακτηριστικών και δεν κάνει εξαντλητική αναζήτηση όπως ο αλγόριθμος Boosting.
- Μπορεί να χρησιμοποιηθεί για ομαδοποίηση.
- Επιτρέπει τη δημιουργία παραλλαγών της βασικής τεχνικής ως προς την κατασκευή του μοντέλου ταξινόμησης π.χ. χρήση διαφορετικών τεχνικών διαμέρισης των παραδειγμάτων των ενδιάμεσων κόμβων.

Τα τυχαία δάση παρουσιάζουν όμως και κάποια σημαντικά μειονεκτήματα ως προς την εφαρμογή τους τα οποία αναφέρονται συνοπτικά παρακάτω:

- Υψηλό υπολογιστικό κόστος.
- Υπάρχει σημαντικό πλήθος ελεύθερων παραμέτρων τις οποίες πρέπει να προσδιορίσει ο χρήστης π.χ. πλήθος δέντρων, βαθμός κόμβων, πλήθος παραδειγμάτων εκπαίδευσης, συνθήκη τερματισμού διαμέρισης των κόμβων.
- Για την επέκταση ενός μοντέλου με στόχο την εισαγωγή μιας ακόμα κατηγορίας απαιτείται η κατασκευή του μοντέλου από την αρχή.
- Κάθε νέο παράδειγμα πρέπει να διασχίσει όλα τα δέντρα του δάσους για την εκτίμηση της κατηγορίας του.

## 2.5 Επεξεργασία Φυσικού Λόγου (Natural Language Processing)

Σε αυτό το σημείο θα γίνει μια εισαγωγή στις τεχνικές επεξεργασίας φυσικού λόγου, καθώς χρησιμοποιούνται ευρέως στο τεχνικό κομμάτι της παρούσας εργασίας. Για να γίνει σωστά η διαδικασία εκπαίδευσης ενός αλγορίθμου μηχανικής μάθησης αρχικά πρέπει να μετατραπούν όλα τα λεκτικά δεδομένα

σε αριθμητικά. Μέχρι να γίνει αυτό, τα δεδομένα περνάνε από μια διαδικασία καθαρισμού θορύβου η οποία θα περιγραφεί σε βήματα παρακάτω.

### 2.5.1 Δειγματοληψία Ενδείξεων (Tokenization)

Tokenization είναι η διαδικασία με την οποία μεγάλη ποσότητα κειμένου χωρίζεται σε μικρότερα τμήματα που ονομάζονται ενδείξεις (tokens). Η επεξεργασία της φυσικής γλώσσας χρησιμοποιείται για την οικοδόμηση εφαρμογών όπως η ταξινόμηση κειμένου, έξυπνα chatbot, συναισθηματική ανάλυση, μετάφραση γλωσσών κλπ. Είναι καθοριστικής σημασίας η κατανόηση του προτύπου στο κείμενο για την επίτευξη του προαναφερθέντος σκοπού. Αυτές οι ενδείξεις είναι πολύ χρήσιμες για την εξεύρεση τέτοιων μοτίβων, καθώς θεωρούνται ως ένα βασικό βήμα για την στελεχοποίηση (stemming) και τη λεμματοποίηση (lemmatization).

Η ουσιαστική διαδικασία του tokenization είναι ότι δημιουργούνται ξεχωριστά tokens για κάθε λέξη μιας πρότασης η ακόμα και σημεία στίξης όπως τελείες, θαυμαστικά κ.α. Ανάλογα με τις ανάγκες του κάθε προβλήματος, το κείμενο μπορεί να χωριστεί και σε μεγαλύτερα tokens πέρα της μίας λέξης η σημείου στίξης κάθε φορά.

### 2.5.2 Στελεχοποίηση (Stemming)

Το Stemming είναι ένα είδος εξομάλυνσης των λέξεων. Η κανονικοποίηση είναι μια τεχνική όπου ένα σύνολο λέξεων σε μια πρόταση μετατρέπεται σε μια ακολουθία για να συντομεύσει την αναζήτηση της. Οι λέξεις που έχουν την ίδια σημασία αλλά έχουν κάποια διακύμανση ανάλογα με το πλαίσιο ή την πρόταση κανονικοποιούνται.

Για κάθε λέξη, υπάρχουν πολλές παραλλαγές της αλλά υπάρχει μια ριζική λέξη. Για παράδειγμα, για την λέξη "eat", η ριζική λέξη είναι "eat" και οι παραλλαγές είναι "eats, eating, eaten κλπ.". Με τον ίδιο τρόπο, με τη βοήθεια του Stemming, μπορεί να βρεθεί η ριζική λέξη οποιασδήποτε παραλλαγής.

**Για παράδειγμα.**

«he was riding»

«he was taking the ride»

Στις δύο παραπάνω προτάσεις, η έννοια είναι η ίδια, δηλαδή η δραστηριότητα ιππασίας στο παρελθόν. Ένας άνθρωπος μπορεί εύκολα να καταλάβει ότι και οι δύο έννοιες είναι οι ίδιες. Αλλά για τις μηχανές, οι δύο προτάσεις είναι διαφορετικές και ως εκ τούτου δύσκολο να μετατραπούν στην ίδια σειρά δεδομένων. Σε περίπτωση που δεν παρέχουμε το ίδιο σύνολο δεδομένων, τότε η μηχανή αποτυγχάνει να προβλέψει. Επομένως, είναι απαραίτητο να μετατραπεί η εκδοχή κάθε λέξης για να προετοιμαστεί το σύνολο δεδομένων

για την εκμάθηση μηχανών. Στην περίπτωση αυτή, η στελεχοποίηση (stemming) χρησιμοποιείται για να ταξινομήσει τον ίδιο τύπο δεδομένων με τη λήψη της ρίζας λέξη «tide».

Ένα απλό παράδειγμα ως προς την χρήση της στελεχοποίησης (stemming) για τις λέξεις “wait, waiting, waited και waits”, δίνεται σε γλώσσα python στο παρακάτω κομμάτι με την αντίστοιχη έξοδο.

```
from nltk.stem import PorterStemmer
e_words = ["wait", "waiting", "waited", "waits"]
ps = PorterStemmer()
for w in e_words:
    rootWord = ps.stem(w)
    print(rootWord)
```

### Output:

```
wait
wait
wait
wait
```

Εικόνα 2-11. παράδειγμα χρήσης της στελεχοποίησης (stemming).

Στο παραπάνω κομμάτι κώδικα φαίνεται μια λίστα με τις διαφορετικές εκδοχές του wait. Μετά την διαδικασία της στελεχοποίησης (stemming) όλες οι εκδοχές κατέληξαν στην ρίζα τους. Δίνεται άλλο ένα παράδειγμα παρακάτω ώστε να κατανοηθεί ακόμα καλύτερα η διαδικασία της δειγματοληψίας ενδείξεων (tokenization) και της στελεχοποίησης (stemming).



```
from nltk.stem import PorterStemmer
from nltk.tokenize import sent_tokenize, word_tokenize
sentence = "Hello sir, You have to build a very good site and I love visiting your site."
words = word_tokenize(sentence)
ps = PorterStemmer()
for w in words:
    rootWord = ps.stem(w)
    print(rootWord)
```

### Output :

```
hello
sir
,
you
have
build
a
veri
good
site
and
I
love
visit
your
site
```

Εικόνα 2-12. Παράδειγμα χρήσης δειγματοληψίας ενδείξεων (tokenization) και στελεχοποίησης (stemming).

Παραπάνω βλέπουμε ότι η παραπάνω πρόταση πέρασε πρώτα την διαδικασία της δειγματοληψίας ενδείξεων (tokenization) ώστε να πάρουμε το κείμενο που έχουμε λέξη-λέξη και στην συνέχεια πέρασε στην διαδικασία της στελεχοποίησης (stemming), ώστε να επιστρέψουν οι ρίζες των εκάστοτε λέξεων.

Εν κατακλείδι, η στελεχοποίηση (stemming), είναι μια διαδικασία προεπεξεργασίας δεδομένων. Η αγγλική γλώσσα έχει πολλές παραλλαγές μιας μόνο λέξης. Αυτές οι παραλλαγές δημιουργούν ασάφεια στην εκπαίδευση και στην πρόβλεψη των μεθόδων μηχανικής μάθησης.

Για να δημιουργήσουμε ένα επιτυχημένο μοντέλο, είναι καθοριστικής σημασίας το φιλτράρισμα αυτών των λέξεων και η μετατροπή στον ίδιο τύπο δεδομένων ακολουθίας (sequence) χρησιμοποιώντας την ρίζα τους. Επίσης, αυτή είναι μια σημαντική τεχνική για τη λήψη δεδομένων σειράς (series data) από μια σειρά προτάσεων και την αφαίρεση περιττών δεδομένων γνωστών και ως κανονικοποίηση (normalization).



### 2.5.3 Λεμματοποίηση (Lemmatization)

Η λεμματοποίηση (Lemmatization) είναι η αλγοριθμική διαδικασία εύρεσης του λήμματος μιας λέξης ανάλογα με το νόημά της. Η λεμματοποίηση συνήθως αναφέρεται στη μορφολογική ανάλυση των λέξεων, η οποία στοχεύει στην απομάκρυνση των λεκτικών καταλήξεων. Βοηθά στην επιστροφή της μορφής βάσης ή λεξικού μιας λέξης, η οποία είναι γνωστή ως το λήμμα. Η μέθοδος NLTK Lemmatization βασίζεται στην ενσωματωμένη λειτουργία `morph` του WorldNet. Το Εργαλείο Φυσικής Γλώσσας (NLTK), είναι μια σειρά από βιβλιοθήκες και προγράμματα για συμβολική και στατιστική επεξεργασία φυσικής γλώσσας για Αγγλικά γραμμένα στη γλώσσα προγραμματισμού Python. Αναπτύχθηκε από τους Steven Bird και Edward Loper στο Τμήμα Επιστήμης Υπολογιστών και Πληροφοριών του Πανεπιστημίου της Πενσυλβανίας. Η προεπεξεργασία κειμένων περιλαμβάνει τόσο την στελεχοποίηση όσο και τη λεμματοποίηση. Η λεμματοποίηση προτιμάται έναντι της στελεχοποίησης (Stemming) λόγω των παρακάτω.

Ο αλγόριθμος στελεχοποίησης (stemming) που περιγράφηκε προηγουμένως, λειτουργεί με το κόψιμο του επιθέματος από τη λέξη. Με μια ευρύτερη έννοια μειώνει είτε την αρχή είτε το τέλος της λέξης.

Αντίθετα, η λεμματοποίηση (Lemmatization) είναι μια πιο ισχυρή πράξη και λαμβάνει υπόψη τη μορφολογική ανάλυση των λέξεων. Επιστρέφει το λήμμα που είναι η βασική μορφή όλων των μορφών του. Απαιτούνται σε βάθος γλωσσικές γνώσεις για τη δημιουργία λεξικών και την αναζήτηση της σωστής μορφής της λέξης. Η στελεχοποίηση (stemming) είναι μια γενική λειτουργία, ενώ η λεμματοποίηση (Lemmatization) είναι μια έξυπνη λειτουργία όπου η σωστή μορφή θα εξεταστεί στο λεξικό. Ως εκ τούτου, η λεμματοποίηση συμβάλλει στη διαμόρφωση καλύτερων χαρακτηριστικών μηχανικής μάθησης.

Παρακάτω παρατίθενται παραδείγματα σε γλώσσα python με σκοπό την σύγκριση των δύο διαδικασιών.

## Stemming code

```
import nltk
from nltk.stem.porter import PorterStemmer
porter_stemmer = PorterStemmer()
text = "studies studying cries cry"
tokenization = nltk.word_tokenize(text)
for w in tokenization:
    print("Stemming for {} is {}".format(w,porter_stemmer.stem(w)))
```

### Output:

```
Stemming for studies is studi
Stemming for studying is studi
Stemming for cries is cri
Stemming for cry is cri
```

## Lemmatization code

```
import nltk
from nltk.stem import WordNetLemmatizer
wordnet_lemmatizer = WordNetLemmatizer()
text = "studies studying cries cry"
tokenization = nltk.word_tokenize(text)
for w in tokenization:
    print("Lemma for {} is {}".format(w, wordnet_lemmatizer.lemmatize(w)))
```

### Output:

```
Lemma for studies is study
Lemma for studying is studying
Lemma for cries is cry
Lemma for cry is cry
```

Εικόνα 2-13. Σύγκριση μεταξύ λεμματοποίησης (Lemmatization) και στελεχοποίησης (stemming).

Βλέπουμε ξεκάθαρα ότι στην περίπτωση της στελεχοποίησης (stemming), ο αλγόριθμος επιστρέφει για τις λέξεις studies και studying την ίδια λέξη πίσω (studi), ενώ στην άλλη περίπτωση της λεμματοποίησης (Lemmatization) ο αλγόριθμος επιστρέφει study στην περίπτωση του studies και studying στην περίπτωση του studying ξεχωρίζοντας έτσι τον παρελθοντικό χρόνο από το παρόν. Επομένως είναι προφανές ότι είναι καλύτερος ένας αλγόριθμος λεμματοποίησης (Lemmatization) σε σχέση με έναν αλγόριθμο στελεχοποίησης (stemming).

Η λεμματοποίηση (Lemmatization) ελαχιστοποιεί την ασάφεια του κειμένου. Παραδείγματα λέξεων όπως ποδήλατο ή ποδήλατα μετατρέπονται σε ποδήλατο. Βασικά, θα μετατρέψει όλες τις λέξεις που έχουν το ίδιο νόημα αλλά διαφορετική αναπαράσταση, στη μορφή της βάσης τους. Μειώνει την πυκνότητα λέξεων στο κείμενο και βοηθά στην προετοιμασία των ενδιαφερόντων χαρακτηριστικών (feature selection) για το μηχανήμα εκπαίδευσης.

Επομένως ως προς τον καθαρισμό των δεδομένων μας, για το πιο έξυπνο και ακριβές μοντέλο εκμάθησης μηχανών η λεμματοποίηση (Lemmatization) είναι η καλύτερη και πιο οικονομική ως προς την μνήμη περίπτωση.

## 2.6 Μη Ισορροπημένα Δεδομένα και Αντιμετώπιση

Ένα από τα πιο συνηθισμένα προβλήματα που πρέπει να αντιμετωπιστεί όταν πρέπει να επιλυθεί κάποιο πρόβλημα μηχανικής μάθησης είναι αυτό των μη ισορροπημένων δεδομένων εκπαίδευσης.

Σε ένα πρόβλημα ταξινόμησης, όταν από όλες τις κλάσεις που θέλουμε να προβλεφθούν, αν για μία ή περισσότερες κλάσεις υπάρχει εξαιρετικά χαμηλός αριθμός δειγμάτων, μπορεί να χρειαστεί να αντιμετωπιστεί το πρόβλημα των μη ισορροπημένων κλάσεων στα δεδομένα μας.

### Παραδείγματα

Πρόβλεψη απάτης (ο αριθμός των απατών θα είναι πολύ χαμηλότερος από τις πραγματικές συναλλαγές)

Προβλέψεις για φυσικές καταστροφές (τα κακά συμβάντα θα είναι πολύ λιγότερα από τα καλά)

Ο εντοπισμός κακοήθους όγκου σε μια ταξινόμηση εικόνων (οι εικόνες με όγκο θα είναι πολύ λιγότερες από αυτές που δεν έχουν όγκο μέσα σε ένα δείγμα εκπαίδευσης)

Αντίστοιχα και στο πρόβλημα μηχανικής μάθησης που αναφέρεται η παρούσα εργασία, έχουμε πολύ λιγότερες περιπτώσεις που υπάρχει στρατολόγηση για τρομοκρατία από ότι περιπτώσεις που υπάρχει απλή ανταλλαγή απόψεων.

Οι μη ισορροπημένες κατηγορίες δημιουργούν ένα πρόβλημα λόγω δύο κύριων θεμάτων:

1. Δεν λαμβάνονται βελτιστοποιημένα αποτελέσματα για την τάξη που είναι ασύμμετρη σε πραγματικό χρόνο, καθώς το μοντέλο / αλγόριθμος δεν λαμβάνει ποτέ επαρκή εικόνα για την υποκείμενη κατηγορία
2. Δημιουργούν πρόβλημα στην διαδικασία της δημιουργίας ενός testing set καθώς λόγω των λιγοστών δειγμάτων που έχουμε για την ασύμμετρη κλάση κατά πάσα πιθανότητα θα έχουμε και λίγα δείγματα για αυτήν στο testing set.

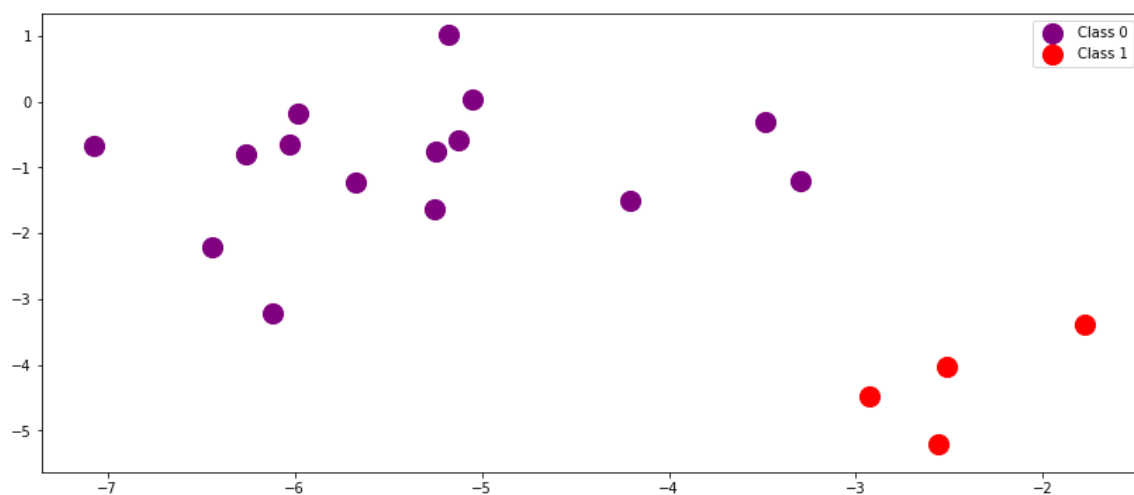
Υπάρχουν τρεις βασικές προσεγγίσεις που προτείνονται, η καθεμιά με τα πλεονεκτήματα και τα μειονεκτήματά της:

1. **Τυχαία Υποδειγματοληψία** - Διαγράφονται τυχαία από την κλάση η οποία έχει επαρκή δείγματα έτσι ώστε η συγκριτική αναλογία των δύο κλάσεων να είναι σημαντική στα δεδομένα μας. Παρόλο που αυτή η προσέγγιση είναι πολύ απλή στην παρακολούθηση, υπάρχει μεγάλη πιθανότητα τα δεδομένα που θα διαγραφούν να περιέχουν σημαντικές πληροφορίες για την κλάση πρόβλεψης.
2. **Τυχαία Υπερδειγματοληψία** - Για την μη ισορροπημένη κλάση αυξάνεται τυχαία ο αριθμός των παρατηρήσεων που είναι απλά αντίγραφα των υπαρχόντων δειγμάτων. Αυτό μας δίνει ιδανικά επαρκή αριθμό δειγμάτων. Η υπερδειγματοληψία όμως μπορεί να οδηγήσει σε υπερφόρτωση (overfit) στα δεδομένα εκπαίδευσης
3. **Συνθετική υπερδειγματοληψία - Συνθετική Τεχνική Υπερδειγματοληψίας Μειονότητας (SMOTE)** - Η τεχνική ζητά να παράγει συνθετικά τις παρατηρήσεις των μη ισορροπημένων τάξεων οι οποίες είναι παρόμοιες με τις υπάρχουσες χρησιμοποιώντας την πλησιέστερη ταξινόμηση των γειτόνων (KNN - K-nearest neighbors).

Στην δική μας περίπτωση χρησιμοποιήθηκε η λύση της συνθετικής υπερδειγματοληψίας καθώς περιγράφεται ευρέως ως η καλύτερη λύση σε προβλήματα λεκτικών δεδομένων.

### 2.6.1 Συνθετική υπερδειγματοληψία - SMOTE (Synthetic Minority Oversampling Technique)

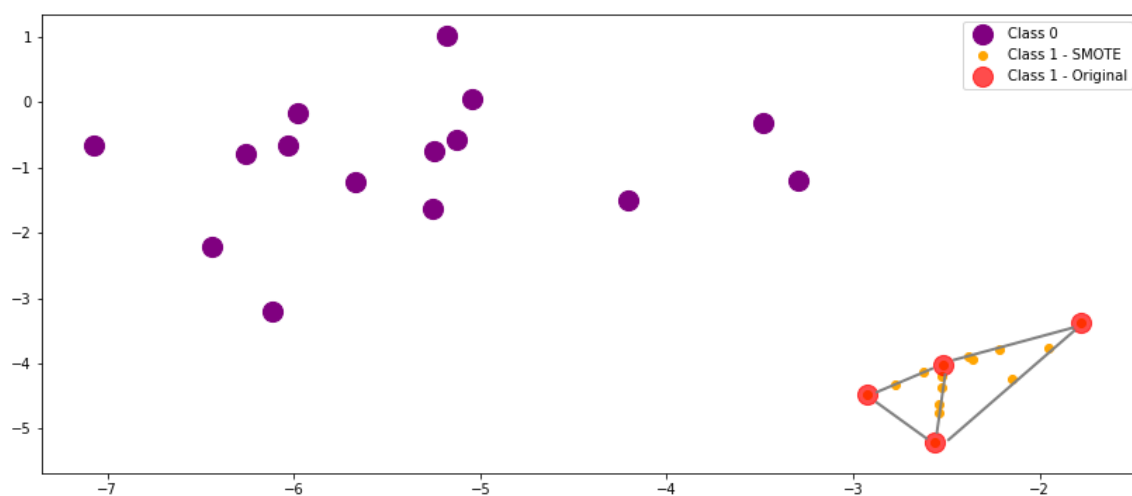
Ακριβώς όπως υποδηλώνει και το όνομα, η τεχνική παράγει συνθετικά δεδομένα για την κλάση που μειονεκτεί. Παρακάτω παρατίθενται μερικές γραφικές αναπαραστάσεις που θα βοηθήσουν στην καλύτερη κατανόηση της διαδικασίας. Παρακάτω βλέπουμε την αρχική αναπαράσταση δύο κλάσεων σε έναν χώρο  $x-y$



Εικόνα 2-14. Αρχική Αναπαράσταση δύο κλάσεων σε έναν χώρο  $x-y$

Με την βοήθεια της συνθετικής υπερδειγματοληψίας παράγονται συνθετικά δεδομένα για την κλάση που μειονεκτεί. Πιο συγκεκριμένα, ο αλγόριθμος συνθετικής υπερδειγματοληψίας - SMOTE λειτουργεί σε 4 απλά βήματα:

1. Επιλέγεται ένα διάνυσμα εισαγωγής της μειονεκτικής κλάσης
2. Βρίσκονται οι  $k$  πλησιέστεροι γείτονες (KNN)
3. Επιλέγεται ένας από αυτούς τους γείτονες και τοποθετείται ένα συνθετικό σημείο οπουδήποτε στη γραμμή που ενώνει το σημείο που εξετάζεται και τον επιλεγμένο γείτονά του
4. Επαναλαμβάνονται τα βήματα μέχρι να εξισορροπηθούν τα δεδομένα



Εικόνα 2-15. Αναπαράσταση δύο κλάσεων σε έναν χώρο x-y μετά την εφαρμογή συνθετικής υπερδειγματοληψίας

Η συνθετική υπερδειγματοληψία προχωρά συνδέοντας τα σημεία της μειονεκτικής κλάσης με τμήματα γραμμών και στη συνέχεια τοποθετεί τεχνητά σημεία σε αυτές τις γραμμές.

## 2.7 Εξαγωγή Χαρακτηριστικών Λεκτικών Δεδομένων

Για προφανείς λόγους ένα μοντέλο μηχανικής μάθησης δεν μπορεί να μάθει από λεκτικά δεδομένα. Επομένως είμαστε αναγκασμένοι να τα μετατρέψουμε με κάποιο τρόπο σε αριθμητικά δεδομένα. Παρακάτω θα αναφερθούν οι κλασικές τεχνικές που χρησιμοποιούνται στον τομέα της μηχανικής μάθησης για την λύση αυτού του προβλήματος.

### 2.7.1 Σάκος Λέξεων - Bag of Words

Ένα από τα πιο κλασσικά μοντέλα για την λύση του προαναφερθέντος προβλήματος.

Το μοντέλο σάκος λέξεων (bag-of-words) είναι μια απλουστευμένη αναπαράσταση που χρησιμοποιείται στην επεξεργασία της φυσικής γλώσσας και στην ανάκτηση πληροφοριών (Information Recovery - IR). Σε αυτό το μοντέλο, ένα κείμενο (όπως μια πρόταση ή ένα έγγραφο) αναπαρίσταται ως η τσάντα (bag)(multiset) των λέξεων της, αγνοώντας τη γραμματική και ακόμη και την τάξη των λέξεων, διατηρώντας όμως την πολλαπλότητα. Περιγράφεται βηματικά η διαδικασία.

1. Εισαγωγή δεδομένων στον αλγόριθμο
2. Μέτρηση της συχνότητας της κάθε λέξης
3. Μετατροπή της κάθε λέξης στην συχνότητα της.

### 2.7.2 TFIDF

Στην ανάκτηση πληροφοριών, το tf-idf ή αλλιώς TFIDF [2] είναι μια αριθμητική στατιστική που προορίζεται να αντικατοπτρίζει τη σημασία μιας λέξης για ένα έγγραφο σε μια συλλογή δεδομένων. Συχνά χρησιμοποιείται ως παράγοντας σταθμίσεως στις αναζητήσεις ανάκτησης πληροφοριών, εξόρυξης κειμένου και μοντελοποίησης χρηστών. Η τιμή tf-idf αυξάνεται αναλογικά με τον αριθμό των φορών που εμφανίζεται μια λέξη στο έγγραφο και αντισταθμίζεται από τον αριθμό των εγγράφων στο σώμα που περιέχουν τη λέξη, γεγονός που βοηθά να προσαρμοστεί για το γεγονός ότι γενικά, ορισμένες λέξεις εμφανίζονται πιο συχνά. Το tf-idf είναι ένα από τα δημοφιλέστερα σχήματα σταθμίσεων σήμερα. Το 83% των συστημάτων αναζήτησης που βασίζονται σε κείμενο στις ψηφιακές βιβλιοθήκες χρησιμοποιούν το tf-idf [2].

Οι παραλλαγές του σχεδίου στάθμισης tf-idf χρησιμοποιούνται συχνά από τις μηχανές αναζήτησης ως κεντρικό εργαλείο για τη βαθμολόγηση και την ταξινόμηση της συνάφειας ενός εγγράφου με δεδομένο ένα ερώτημα χρήστη. Το tf-idf μπορεί να χρησιμοποιηθεί με επιτυχία για φιλτράρισμα των λέξεων σταματήματος (stopwords) σε διάφορα πεδία δειγμάτων, συμπεριλαμβανομένης της περιλήψης κειμένου και της ταξινόμησης. Μια από τις απλούστερες λειτουργίες κατάταξης υπολογίζεται με άθροιση του tf-idf για κάθε όρο. Πολλές πιο εξελιγμένες λειτουργίες κατάταξης είναι παραλλαγές αυτού του απλού μοντέλου.

Τα δύο παραπάνω μοντέλα είναι τα πιο διάσημα ως προς την εξαγωγή χαρακτηριστικών από λεκτικά δεδομένα. Στην περίπτωση μας χρησιμοποιήθηκε το bag-of-words λόγω των καλύτερων αποτελεσμάτων που εξάγει, καθώς είναι αποδεδειγμένο ότι λειτουργεί πολύ καλύτερα για δεδομένα μικρού όγκου όπως είναι τα δικά μας.

### 2.7.3 TFIDF vs Bag of words

Παρά την γενικότερη πεποίθηση ότι η τεχνική Bag of Words υπερτερεί της τεχνικής TFIDF, ειδικότερα, όταν τα δεδομένα είναι λίγα όπως στην δική μας περίπτωση, κρίθηκε σκόπιμο να γίνει μια σύγκριση μεταξύ των δύο τεχνικών ώστε να καταλήξουμε σε κάποιο συμπέρασμα για το ποια από τις δύο θα χρησιμοποιήσουμε.

Η σύγκριση έγινε με την χρήση ενός απλού μοντέλου συσχέτισης cross correlation ώστε να δούμε ποια από τα παραγόμενα tokens των δύο τεχνικών δημιουργούν καλύτερες συσχετίσεις. Όλα τα παραπάνω συμπεριλαμβάνονται στον κώδικα που βρίσκεται στο παράρτημα. Με βάση τα αποτελέσματα, καταλήξαμε στο ότι για το πρόβλημα μας, η τεχνική bag of words είναι πιο αποδοτική σε σχέση με την τεχνική TFIDF.





### 3 ΑΝΑΛΥΣΗ ΚΑΙ ΛΥΣΗ ΤΟΥ ΠΡΟΒΛΗΜΑΤΟΣ ΜΑΣ.

Οι προτεραιότητες στην καταπολέμηση της διεθνούς τρομοκρατίας έχουν επεκταθεί πέραν της καταστολής της δράσης των τρομοκρατικών οργανώσεων, στην ανάδειξη ζητημάτων πρόληψης της τρομοκρατικής απειλής, όπως η αποτροπή της ριζοσπαστικοποίησης και στρατολόγησης νέων μαχητών.

Οι πρόσφατες τρομοκρατικές επιθέσεις στο έδαφος της ΕΕ έχουν καταδείξει τον αθέμιτο τρόπο με τον οποίο οι τρομοκράτες χρησιμοποιούν τα μέσα κοινωνικής δικτύωσης, ώστε να στρατολογούν υποστηρικτές και να τους προετοιμάζουν. Οι τρομοκράτες χρησιμοποιούν κρυπτογραφημένες μεθόδους επικοινωνίας για να σχεδιάζουν και να υποβοηθούν τις τρομοκρατικές τους δραστηριότητες. Επίσης χρησιμοποιούν το διαδίκτυο για να εξυμνούν τις πράξεις τους, να ενθαρρύνουν και άλλους να ακολουθήσουν αυτό το παράδειγμα και να προξενήσουν τον φόβο στο ευρύ κοινό.

Το τρομοκρατικό περιεχόμενο που κοινοποιείται στο διαδίκτυο για τους σκοπούς αυτούς διαδίδεται μέσω παρόχων υπηρεσιών φιλοξενίας δεδομένων που επιτρέπουν την ανάρτηση περιεχομένου τρίτων. Όπως αποδείχθηκε σε αρκετές πρόσφατες τρομοκρατικές επιθέσεις εντός της Ευρώπης, το τρομοκρατικό περιεχόμενο στο διαδίκτυο επιδρά καθοριστικά στη ριζοσπαστικοποίηση ατόμων και ομάδων και την υποκίνηση επιθέσεων.

Σε αυτό το κεφάλαιο θα περιγραφούν πιο συγκεκριμένα οι διαδικασίες και οι αναλύσεις που έγιναν ώστε να φτάσουμε στην βέλτιστη λύση του προβλήματος μας, που είναι η πρόβλεψη στρατολόγησης τρομοκρατών και συνεπώς τρομοκρατικών επιθέσεων με τη χρήση τεχνικών μηχανικής μάθησης.

Στην πράξη, θα εφαρμοστούν οι μηχανισμοί επεξεργασίας φυσικού λόγου και οι αλγόριθμοι μηχανικής μάθησης που αναλύθηκαν στο προηγούμενο κεφάλαιο και θα συγκριθούν μεταξύ τους ως προς την αποτελεσματικότητά τους. Συμπληρωματικά θα αποδειχθεί η υπεροχή της τεχνικής Bag of Words έναντι της τεχνικής TFIDF και η συμβολή της τεχνικής της συνθετικής δειγματοληψίας στην εξαγωγή βελτιστοποιημένων αποτελεσμάτων. Ο στόχος είναι η αναγνώριση λεκτικών προτύπων από αναρτήσεις ιστολογίου (blog posts) και φόρουμ του σκοτεινού διαδικτύου (dark web), ώστε να βρεθεί η συσχέτιση των λέξεων ως προς το αν αυτά απευθύνονται σε τρομοκρατική στρατολόγηση ή όχι.

Τα δεδομένα που χρησιμοποιήθηκαν στην παρούσα εργασία συλλέχθηκαν από φόρουμ από το σκοτεινό διαδίκτυο (Dark Web). Το αρχείο δεδομένων είναι συμβατό με αρχεία τιμών διαχωρισμένων με κόμματα (Comma-Separated Values - CSV) .

Τα δεδομένα αυτά θεωρούνται αξιόπιστα καθώς έχουν προκύψει από ετυμηγορία πραγματογνωμόνων, οι οποίοι εξέτασαν τα μηνύματα και συμφώνησαν μεταξύ τους αν σε κάποια από τα μηνύματα διαφαίνεται στρατολόγηση τρομοκρατών ή όχι.

Το δείγμα μας είναι αρκετά μικρό, μόλις 295 εγγραφές. Κάποιες από αυτές έχουν μεταφραστεί στα Αγγλικά από άλλη γλώσσα κατά τη διαδικασία αξιολόγησης. Παρόλα αυτά θα φανεί παρακάτω ότι είναι αρκετό για να μας δώσει πολύ ικανοποιητικά αποτελέσματα, όταν χρησιμοποιηθούν οι κατάλληλες τεχνικές προετοιμασίας δεδομένων και επιλογής μοντέλου.

Τα δεδομένα που χρησιμοποιήθηκαν βρίσκονται στο αρχείο Ansar1\_original.xlsx, παρακάτω φαίνεται ένα δείγμα της αρχικής μορφής των δεδομένων μέσα από την εφαρμογή Microsoft Excel.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1				JacobRaw												
2	AgreeID	MessageID	ThreadID	Recruitment	Translation	Message										
3	1	31599	11251	0		{Source: Digital Chicago, Inc} North Side resident Uzair Ali Hashmi informed the FBI last month that terrorists w...										
4	2	117	98	1		In the name of God the Merciful Praise be to Allah, and enough is enough, prayer and peace be upon His Proph...										
5	3	1606	520	0		Did Mansoor join the emerat? I heard he is still fighting for ichkira Republic										
6	4	5644	188	0		I dont think 'juba' is one person. from iai site Quote: Q: Baghdad Sniper is a symbol and one of the heroes of i...										
7	5	17598	6646	0		CIA recruiting Arabic, Farsi, Urdu speakers Sat, 30 May 2009 21:13:34 GMT The Central Intelligence Agency (CIA) ...										
8	6	1788	1032	0		Quote: US plan to arm militias scares some in Afghanistan By KATHY GANNON - 7 hours ago MAIDAN SHAHR, Af...										
9	7	18529	7013	0		may Allah (s.w) give them Jannatul-Firdaws... and accept them as Shuhadaas.										
10	8	22654	8253	0		Assalamu'alaikum When I read this, I thought that the madrasahs in Thailand are teaching young Muslims wel...										
11	9	22763	8157	0		May Allah guard them and accept them into Jannat al-Firdaws.										
12	10	43628	14541	0		Al Shabaab takes control of north Mogadishu MOGADISHU (Mareeg) — Al Shabaab militants have taken control o...										
13	11	25956	9400	0		(IsraelNN.com) A court in the German city of Koblenz sentenced a German of Pakistani origin to eight years in p...										
14	12	34231	12032	0		Salaam Alaykom, Quote: Russia objects to the attribution of Islamic affiliations to extremists hailing the Muslim...										
15	13	45048	14760	0		AssalamAlaikum Brothers inshaAllah these are all lies from the Headquarters of the Pakistan Army, Mulla Nazir a...										
16	14	48145	15728	0		Street-to-street fighting rages in key Taliban base Islamabad: The Pakistani Army said on Wednesday its forces l...										
17	15	49002	15812	0		Quote: Originally Posted by TheRealTruth BLA is a secular unislamic nationalist group and it is feared that of the...										
18	16	13541	5329	1		"A Golden chance to Join Jihad in Somalia" was the title of a message launched today on al Falojah forums by a i...										
19	17	14764	5680	0		Quote: Originally Posted by abu baraa ameen, so was there any responses about how to go? Akhil, i think its jus...										
20	18	26340	9529	0		Got what she deserved. Quote: A woman was jailed for four years after blackmailing her Muslim friend by threa...										
21	19	49698	15840	0		brothers in the middle east there are jews, christians and their equal atheists. shia is muslims who supported a...										
22	20	25994	9413	0		BERLIN/KOBLENZ: A state court on Monday convicted a German man of Pakistani origin for helping to fund and s...										
23	21	35305	12360	1		By Noor Ali Wed Sep 2, 2009 GARISSA, Kenya, Sept 2 (Reuters) - Chaos in Somalia is spilling over its borders, fue...										
24	22	42558	19599	0		IMMEDIATE RELEASE No. 763-09 October 01, 2009 DoD Identifies Army Casualties The Department of Defense an...										
25	23	41848	14083	0		Apostates recognize that young people go to the mountains Publication time: Today at 12:04 of Jowhar Police pr...										
26	24	46930	14479	0		double post										
27	25	6163	2717	1		Somali terror group raps in English for recruits WASHINGTON (AFP) — A propaganda video by Islamic extremists...										
28	26	16245	6169	1		US intelligence: more than 1000 foreign fighters involve in Somali fighting Posted: 5/24/2009 6:13:00 PM Shabell										
29	27	17072	6366	0		Quote: Originally Posted by Thunderman I have now added him as a friend on Facebook. But something tells m...										
30	28	20460	7592	0	Translated	Glory be to God, His Prophet and the believers Information Office of the Ansar al-Sunna (the Shariah) 20 / Inan...										
31	29	22399	7802	0		inshallah, these kuffars will dont get victory. and they will defeat soon. we must cut these puppets head. and sl...										
32	30	59488	17537	0		Quote: Originally Posted by Al-ibin Brother Umar I love you for the sake of Allah. Me too.										
33	31	13045	5140	0		May Allah make it easy on th captives...										
34	32	22650	7805	0		Assalamu'alaikum May Allah (SWT) reward our brothers for exposing this plot and so that other brothers and i...										
35	33	35036	12216	0		Sallams Like he didn't know it going on at the time Do they think we're stupid!!!										
36	34	36048	12585	0		American Islamist Killed as Somali Clashes Intensify Written by Benjamin Joffe-Walt Published Sunday, Septem...										
37	35	44840	14830	0		9:57. The Arabs of the desert are the worst in Unbelief and hypocrisy, and most fitted to be in ignorance of the c...										
38	36	20930	7644	0		Quote: Originally Posted by FatimaLaRose [Source: NEWKERALA.COM ] Terrorists using children for suicide atta...										
39	37	24610	8727	0		I do have experience with Jaish-e-Muhammad Memebbers. They were good Mujahideen and actively fought agai...										
40	38	24613	8885	0		Usually it is the gov't demanding the rebels to lay down their arms! Lol! May God grant them Mujahideen victor...										
41	39	34912	11951	0		Quote: Originally Posted by Thunderman Why not? I think all muslim countries have their national teams and e...										
42	40	16312	6254	0		Brits use SEO strategies to fight terrorism http://news.cnet.com/8301-13639_3-10223182-42.html Islam is getting										

Εικόνα 3-1. δείγμα της αρχικής μορφής των δεδομένων μέσα από την εφαρμογή Microsoft Excel

Και πως αυτά αντίστοιχα εμφανίζονται όταν εισαχθούν προς περαιτέρω επεξεργασία

	Unnamed: 0	Unnamed: 1	Unnamed: 2	JacobRaw	Unnamed: 4	Unnamed: 5
0	AgreeID	MessageID	ThreadID	Recruitment	Translation	Message
1	1	31599	11251	0	NaN	{Source: Digital Chicago, Inc} North Side resi...
2	2	117	98	1	NaN	In the name of God the Merciful Praise be to A...
3	3	1606	520	0	NaN	Did Mansoor join the emerat? I heard he is stil...
4	4	5644	188	0	NaN	I dont think 'juba' is one person. from iai s...

Εικόνα 3-2. δείγμα της αρχικής μορφής των δεδομένων που μόλις έχουν εισαχθεί προς επεξεργασία

Βλέπουμε ότι υπάρχουν 6 στήλες (columns) όπου οι πιο σημαντικές είναι η «Recruitment» που ουσιαστικά λειτουργεί και ως μεταβλητή – στόχος (target variable) και η στήλη «Message» που περιέχει τα λεκτικά χαρακτηριστικά του εκάστοτε μηνύματος. Το «Recruitment» παίρνει δύο τιμές, 0 για Not Recruitment (μη στρατολόγηση) και 1 για Recruitment (στρατολόγηση).

### 3.1 Διαδικασία Προεπεξεργασίας Λεκτικών Δεδομένων

Η πρώτη διαδικασία που έγινε ήταν η λεκτική ανάλυση και όλο το κομμάτι τις προεπεξεργασίας των λεκτικών δεδομένων έτσι όπως αναλύθηκε στο κεφάλαιο 2.5. Παρακάτω περιγράφονται τα βήματα.

- Μετατροπή όλων των γραμμών σε πεζά
- Αφαίρεση των τελειών, θαυμαστικών και γενικά σε οτιδήποτε έχει να κάνει με punctuation, καθώς επιπροσθέτουν επιπλέον αριθμητικές διαστάσεις οι οποίες δεν είναι χρήσιμες για την λύση του προβλήματος
- Αφαίρεση των δεδομένων που δεν είναι αλφαβητικά για τον ίδιο λόγο που έγινε και το βήμα 2
- Αφαίρεση των κενών
- Χρήση της λεμματοποίησης (Lemmatization) στα δεδομένα μας (αναλύθηκε στο κεφάλαιο 2.5.3 για ποιο λόγο έγινε χρήση λεμματοποίησης (Lemmatization) αντί για στελεχοποίηση (stemming)).

Στην παρακάτω εικόνα (**Σφάλμα! Το αρχείο προέλευσης της αναφοράς δεν βρέθηκε.**3), βλέπουμε την εκτελέσιμη διαδικασία.

```
from nltk.tokenize import word_tokenize
import string
from nltk.corpus import stopwords
from nltk.stem import WordNetLemmatizer
from nltk.stem.porter import PorterStemmer

text = []
for message in data['Unnamed: 6'].values[1:]:
    text.append([message])

def text_proc(data):
    for message in range(len(data)):
        data[message] = word_tokenize(data[message][0])
        # convert to lower case
        data[message] = [w.lower() for w in data[message]]
        # remove punctuation from each word
        table = str.maketrans('', '', string.punctuation)
        data[message] = [w.translate(table) for w in data[message]]
        # remove remaining tokens that are not alphabetic
        words_n1 = [word for word in data[message] if word.isalpha()]
        # filter out stop words whitespaces etc.
        stop_words = set(stopwords.words('english'))
        data[message] = [w for w in words_n1 if not w in stop_words]
        lemmatizer = WordNetLemmatizer() # use the porter stemmer to stem all the words
        data[message] = [lemmatizer.lemmatize(word) for word in data[message]]
    return data

text = text_proc(text)
print(text[0:2]) # print two samples 0 to 2:

labels = data['JacobRaw'].values.tolist()[1:]
labels = np.array(labels)

[['source', 'digital', 'chicago', 'inc', 'north', 'side', 'resident', 'uzair', 'ali', 'hashmi', 'informed', 'fbi', 'last', 'mon
th', 'terrorist', 'trying', 'recruit', 'one', 'gave', 'letter', 'stating', 'job', 'carry', 'mission', 'giving', 'nonbeliever',
'deserve', 'hashmi', 'said', 'letter', 'also', 'asked', 'familiar', 'downtown', 'chicago', 'stated', 'calling', 'jihad', 'broth
er', 'key', 'role', 'operation', 'investigation', 'found', 'hashmi', 'made', 'story', 'fabricated', 'letter', 'authority', 'sai
d', 'wednesday', 'indicted', 'federal', 'grand', 'jury', 'three', 'count', 'making', 'false', 'statement', 'fbi', 'according',
'indictment', 'hashmi', 'contacted', 'fbi', 'july', 'saying', 'someone', 'approached', 'previous', 'day', 'asked', 'proficien
t', 'firearm', 'suggested', 'join', 'god', 'military', 'person', 'hashmi', 'claimed', 'approached', 'actually', 'made', 'statem
ent', 'indictment', 'said', 'take', 'seriously', 'allegation', 'terrorism', 'activity', 'aggressively', 'investigate', 'every',
'lead', 'said', 'robert', 'grant', 'special', 'agentincharge', 'chicago', 'fbi', 'want', 'encourage', 'people', 'report', 'genu
inely', 'suspicious', 'activity', 'also', 'seek', 'prosecute', 'anyone', 'deliberately', 'provides', 'false', 'information', 'd
iverts', 'agent', 'resource', 'important', 'matter', 'convicted', 'hashmi', 'could', 'get', 'year', 'prison', 'fine', 'schedule
d', 'arraigned', 'wednesday', 'u', 'district', 'court'], ['name', 'god', 'merciful', 'praise', 'allah', 'enough', 'enough', 'pr
ayer', 'peace', 'upon', 'prophet', 'chosen', 'ansar', 'almujahideen', 'progress', 'incitement', 'u', 'people', 'sunni', 'commun
ity', 'lebanon', 'join', 'fatah', 'alislam', 'supporting', 'need', 'thanks', 'god', 'almighty', 'produce', 'version', 'voicepro
vocation', 'incite', 'young', 'sunni', 'brave', 'lebanon', 'god', 'brother', 'fatah', 'alislam', 'defend', 'support', 'despit
e', 'tkhalakn', 'epic', 'nahr', 'albare', 'taidoa', 'sin', 'may', 'god', 'forgive', 'u', 'every', 'nation', 'unification', 've
rsion', 'new', 'voice', 'occasion', 'eid', 'sunni', 'lebanon', 'download', 'file', 'size', 'high', 'quality', 'mb', 'forget',
'mujahideen', 'mesh', 'مَش', 'fit', 'ansar', 'almujahideen']]
```

Εικόνα 3-3. Διαδικασία Προεπεξεργασίας Λεκτικών Δεδομένων

Μετά από αυτήν την διαδικασία τα δεδομένα μας έχουν μετατραπεί σε μια καλύτερη και αξιοποιήσιμη μορφή για ένα μοντέλο μηχανικής μάθησης πέρα από το γεγονός ότι δεν έχουν έρθει ακόμα σε αριθμητική μορφή.

Παρακάτω δίνεται μια πιο αναλυτική επεξήγηση της διαδικασίας που ακολουθήθηκε.

1. Τα ανεπεξέργαστα δεδομένα (raw data) περνάνε από δειγματοληψία (tokenizer) ώστε να μετατραπούν σε word tokens.
2. Μετατροπή όλων των λέξεων σε πεζά (lower case). Εισάγουμε τα αρχικά tokens και παίρνουμε πίσω τα tokens σε πεζά (lower case tokens).
3. Αφαιρούμε τον θόρυβο αφαιρώντας θαυμαστικά, τελείες κλπ.



4. Τα ήδη «καθαρισμένα» δεδομένα (curated data) περνάνε μέσα από διαδικασία λεμματοποίησης (lemmatizer) ώστε να γίνει η μείωση διαστάσεων (dimensionality reduction) που συζητήθηκε στο προηγούμενο κεφάλαιο.

Επόμενο βήμα είναι η μετατροπή των λεκτικών δεδομένων σε αριθμητικά δεδομένα. Για αυτήν την διαδικασία χρησιμοποιήθηκε ο αλγόριθμος του Bag-Of-Words λόγω των καλύτερων αποτελεσμάτων που εξάγει, για δεδομένα μικρού όγκου σε σχέση με τον αλγόριθμο TFIDF. Αυτό θα αποδειχθεί πρακτικά παρακάτω όπου εξετάζουμε την αποδοτικότητα τους όταν εφαρμόζονται σε συνδυασμό με τον αλγόριθμο Random Forest.

Παρακάτω βλέπουμε την εκτελέσιμη διαδικασία.

```
: join_text = []
for message in range(len(text)):
    join_text.append(' '.join(text[message])) # join all the text to feed it into the bag of words.

import collections, re
# create the bagofwords
bags_of_words = [collections.Counter(re.findall(r'\w+', txt)) for txt in join_text]
sumbags = sum(bags_of_words, collections.Counter())

: # transform all the words from our features matrix into frequency numerics
for message in text:
    for i in range(len(message)):
        message[i] = sumbags[str(message[i])]

: print(text[0:2]) # check if the transform was successful

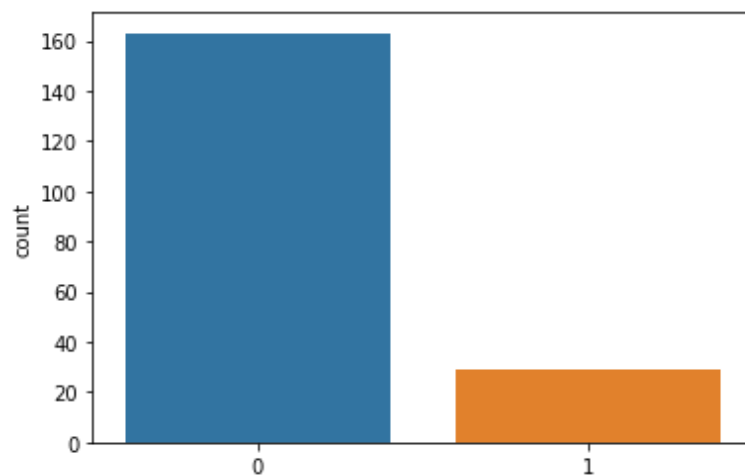
[[55, 2, 4, 1, 43, 19, 10, 1, 17, 6, 6, 33, 50, 69, 73, 25, 70, 150, 16, 13, 1, 10, 18, 8, 12, 2, 4, 6, 502, 13, 166, 26, 4, 3,
4, 11, 9, 132, 83, 18, 10, 53, 23, 18, 6, 32, 10, 1, 13, 64, 502, 18, 2, 19, 3, 2, 52, 9, 16, 6, 29, 33, 66, 13, 6, 2, 33, 17,
34, 18, 5, 4, 54, 26, 2, 3, 5, 131, 30, 91, 30, 6, 17, 5, 7, 32, 29, 13, 502, 50, 5, 3, 46, 17, 2, 2, 38, 8, 502, 5, 10, 25, 1,
4, 33, 44, 7, 174, 49, 1, 4, 17, 166, 17, 1, 16, 2, 2, 6, 24, 1, 17, 10, 14, 8, 10, 6, 41, 42, 127, 30, 1, 3, 1, 18, 287, 25, 4,
2], [23, 30, 5, 13, 130, 13, 13, 10, 25, 38, 7, 3, 4, 2, 6, 1, 287, 174, 13, 37, 6, 131, 2, 3, 14, 33, 3, 30, 6, 5, 5, 1, 1, 5,
5, 13, 1, 6, 30, 83, 2, 3, 11, 65, 22, 1, 1, 1, 1, 1, 3, 107, 30, 2, 287, 38, 31, 1, 5, 77, 5, 3, 1, 13, 6, 3, 1, 5, 15, 2, 7,
2, 77, 1, 1, 2, 4, 2]]

: text = np.array(text)
seq_len = 1100
features = np.zeros((len(text), seq_len), dtype=int)
for i, row in enumerate(text):
    features[i, -len(row):] = np.array(row)[:seq_len]
```

Εικόνα 3-4. Διαδικασία αλγορίθμου Bag-Of-Words

Ένα από τα πιο σημαντικά βήματα που έγινε για να καταλήξουμε στα βέλτιστα αποτελέσματα είναι το feature engineering που έγινε πάνω στα δεδομένα μας. Για να καταλήξουμε στο ότι πρέπει να γίνει κάτι τέτοιο έγινε μια μικρή ανάλυση στα δείγματά μας (samples).

Στο παρακάτω plot βλέπουμε τον αριθμό των δειγμάτων (samples) που υπάρχουν σε κάθε διαφορετική κλάση.



Εικόνα 3-5. Countplot των δειγμάτων μας πριν την εφαρμογή συνθετικής υπερδειγματοληψίας

Όπως βλέπουμε παραπάνω υπάρχουν πάνω από 160 δείγματα (samples) που είναι μηνύματα «Not Recruitment» και σχεδόν 40 samples που είναι «Recruitment». Οπότε αμέσως γίνεται κατανοητό ότι το Recruitment μειονεκτεί απέναντι στο Not Recruitment. Έτσι προκύπτει ότι πρέπει επιπρόσθετα να λυθεί και ένα πρόβλημα μη ισορροπημένων δεδομένων για να καταλήξουμε στην βέλτιστη λύση. Χρησιμοποιήθηκε η συνθετική υπερδειγματοληψία - SMOTE που περιεγράφηκε στο κεφάλαιο 2.6.

Μετά από την διαδικασία συνθετικής υπερδειγματοληψίας - SMOTE δημιουργήθηκαν επιπλέον συνθετικά δείγματα (samples) για την βελτιστοποίηση της διαδικασίας εκμάθησης. Παρακάτω φαίνεται ο αριθμός των δειγμάτων (samples) μετά την χρήση της συνθετικής υπερδειγματοληψίας - SMOTE. Επίσης δίνεται και ο εκτελέσιμος κώδικας.

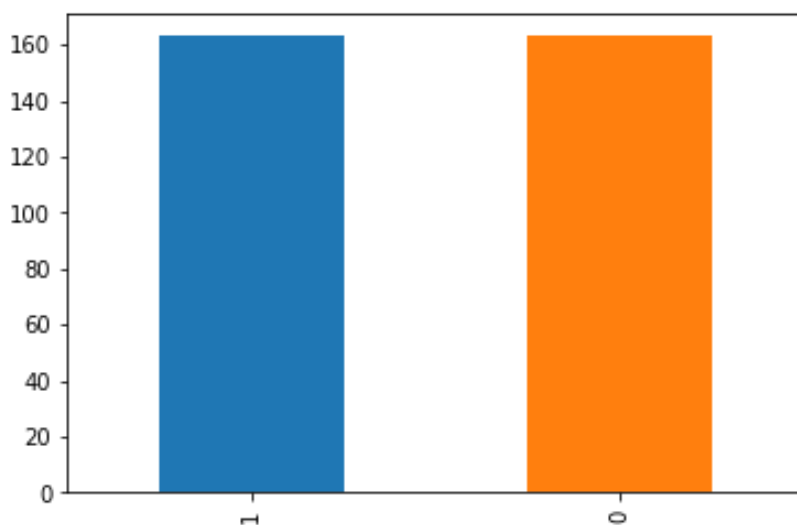
```
from imblearn.over_sampling import SMOTE
from sklearn.model_selection import train_test_split

sm = SMOTE(random_state=33)
X_train_new, y_train_new = sm.fit_sample(features, labels.ravel())

train_x, test_x, train_y, test_y = train_test_split(X_train_new, y_train_new, test_size=0.2, random_state=100)

pd.Series(y_train_new).value_counts().plot.bar()
```

Εικόνα 3-6. διαδικασία συνθετικής υπερδειγματοληψίας - SMOTE



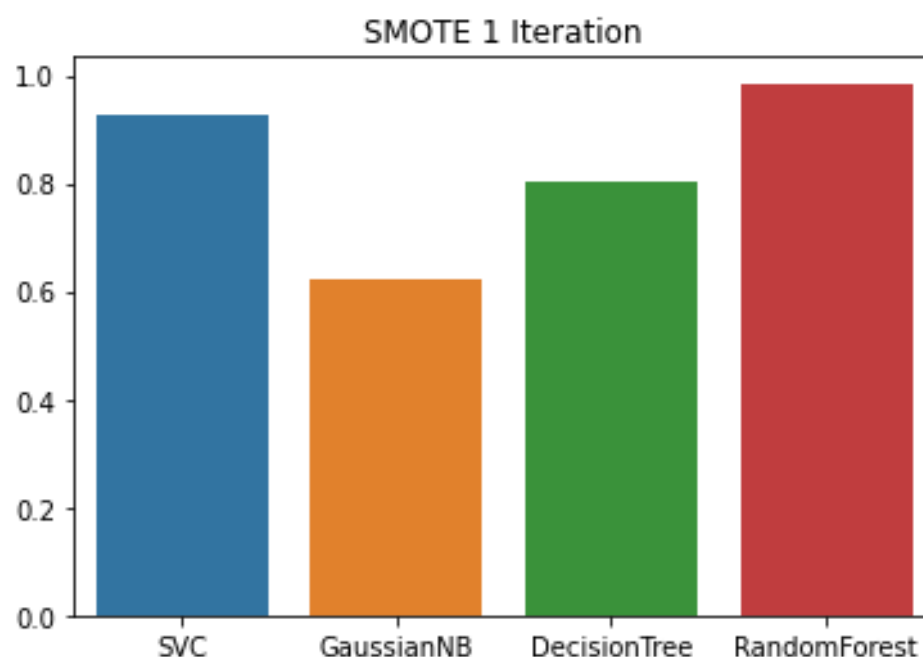
Εικόνα 3-7. Countplot των δειγμάτων μας μετά την εφαρμογή συνθετικής υπερδειγματοληψίας

Μετά από την εκτέλεση της διαδικασίας συνθετικής υπερδειγματοληψίας – SMOTE, φαίνεται ξεκάθαρα ότι τα δεδομένα μας είναι πλέον ισορροπημένα.

### 3.2 Διαδικασία Εκμάθησης των Μοντέλων μας.

Μετά την διαδικασία προεπεξεργασίας έρχεται η διαδικασία εκμάθησης των μοντέλων μας. Αρχικά, κρατήθηκε το 80% των δεδομένων για την εκπαίδευση των μοντέλων και το 20% για τον έλεγχο της απόδοσης των εκπαιδευμένων μοντέλων πάνω σε δεδομένα τα οποία δεν έχουν ξαναδεί. Χρησιμοποιήθηκαν τα μοντέλα μηχανικής μάθησης **Support Vector Machines (Support Vector Classifier – SVC)**, **Gaussian Naïve Bayes**, **Decision Tree** και **Random Forest**. Τονίζεται πως χρησιμοποιήθηκε τυχαία υπερδειγματοληψία (random seed) στο μοντέλο, το οποίο σημαίνει πως κάθε φορά που εκτελείται η διαδικασία εκπαίδευσης επιλέγεται διαφορετικό υπόδειγμα για τα δύο δείγματα που έχουμε. Στην παρακάτω εικόνα φαίνεται η απόδοση του κάθε μοντέλου ξεχωριστά.

Θα ξεκινήσουμε την εκτέλεση με την εφαρμογή συνθετικής υπερδειγματοληψίας SMOTE



SVC	Gaussian NB	Decision Tree	Random Forest
92.42%	62.12%	77.27%	98.48%

Εικόνα 3-8. Barplot αποτελεσμάτων (Με χρήση συνθετικής υπερδειγματοληψίας, 1 Επανάληψη).

Παρατηρούμε ότι τα τυχαία δάση (Random Forests), στην περίπτωση όπου εφαρμόζεται συνθετική υπερδειγματοληψία SMOTE ανταποκρίνονται πολύ καλά στο πρόβλημά μας, με απόδοση 98,48%. Πολύ καλά επίσης ανταποκρίνεται και ο αλγόριθμος Support Vector Machines (SVC) με απόδοση 92,42%, ενώ οι αλγόριθμοι Decision Tree με απόδοση 77,27% και Gaussian NB με απόδοση 62,12% δεν δίνουν ιδιαίτερα ικανοποιητικά αποτελέσματα. Παρακάτω δίνεται και ο αντίστοιχος εκτελέσιμος κώδικας,



```
from sklearn.svm import SVC
from sklearn.naive_bayes import GaussianNB
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import train_test_split
from sklearn.model_selection import cross_validate
from sklearn.ensemble import RandomForestClassifier

clf = SVC(kernel='poly',tol=3e-2,C=10, gamma="auto").fit(train_x, train_y)
clf1 = GaussianNB().fit(train_x, train_y)
clf2 = DecisionTreeClassifier().fit(train_x, train_y)
clf3 = RandomForestClassifier().fit(train_x, train_y)
clf4 = RandomForestClassifier().fit(x_train_tfidf, y_train_tfidf)
score = clf.score(test_x, test_y)
score1 = clf1.score(test_x, test_y)
score2 = clf2.score(test_x, test_y)
score3 = clf3.score(test_x, test_y)
score4 = clf4.score(x_test_tfidf, y_test_tfidf)

print(score)
print(score1)
print(score2)
print(score3)
print("tfidf score with Random Forest " + str(score4))

y = [score, score1, score2, score3]
x = ["SVC", "GaussianNB", "DecisionTree", "RandomForest"]
# if the scores are changing it's rational since almost everytime we run this a different part of the data to validate.

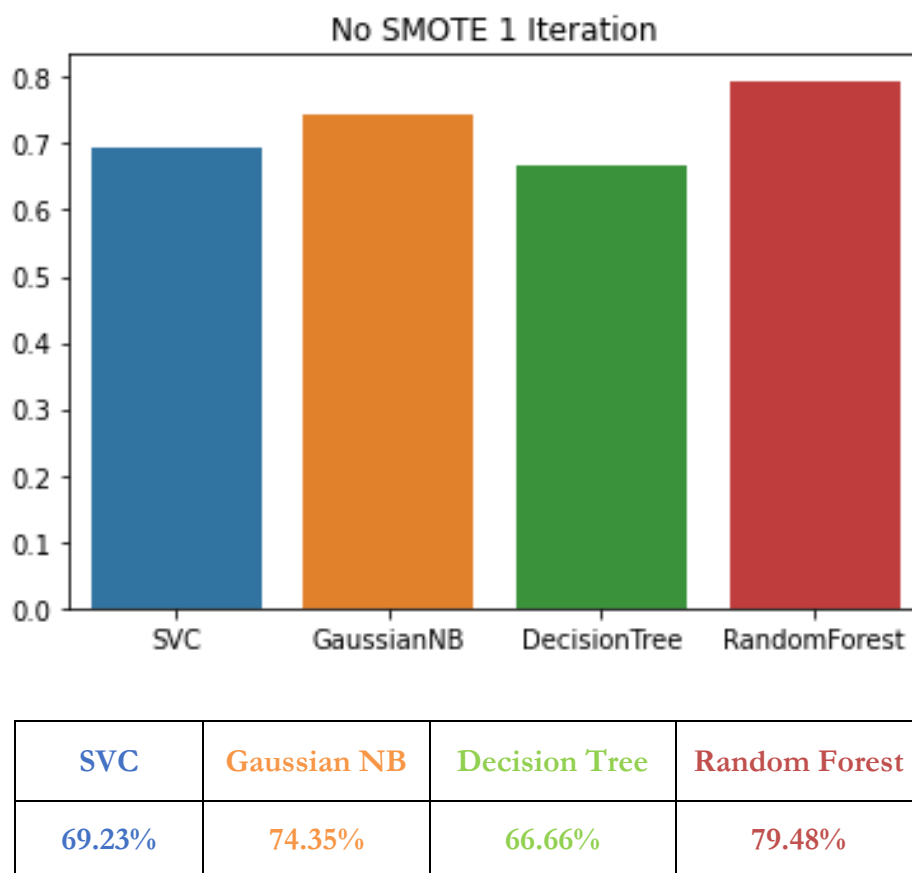
sns.barplot(x=x, y=y).set_title('SMOTE 1 Iteration')

0.9242424242424242
0.6212121212121212
0.7727272727272727
0.9848484848484849
tfidf score with Random Forest 0.8205128205128205
```

Εικόνα 3-9. Εκτελέσιμος κώδικας μοντέλων (Με χρήση συνθετικής υπερδειγματοληψίας, 1 Επανάληψη).

Σε αυτό το σημείο εξετάζεται πόσο σημαντική είναι επιλογή της μεθόδου εξαγωγής χαρακτηριστικών λεκτικών δεδομένων στην τελική απόδοση του μοντέλου και συγκεκριμένα, η συνεισφορά του αλγορίθμου Bag-Of-Words σε σχέση με τον αλγόριθμο TFIDF στην αύξηση της τελικής απόδοσης του αλγορίθμου Random Forest. Είναι ξεκάθαρο ότι ο αλγόριθμος Bag-Of-Words υπερέχει ξεκάθαρα με απόδοση 98,48% σε σχέση με τον αλγόριθμο TFIDF όπου η απόδοση είναι μόλις 82,05%.

Παρακάτω παρατίθενται τα αποτελέσματα χωρίς την χρήση συνθετικής υπερδειγματοληψίας SMOTE ώστε να μπορούμε να τα συγκρίνουμε με τα αντίστοιχα όπου εφαρμόζεται η συνθετική υπερδειγματοληψία.



Εικόνα 3-10. Barplot αποτελεσμάτων (Χωρίς χρήση συνθετικής υπερδειγματοληψίας, 1 Επανάληψη).

Εδώ φαίνεται πόσο καθοριστική είναι η επιρροή της συνθετικής υπερδειγματοληψίας στην απόδοση των αλγορίθμων. Στην περίπτωση που δεν γίνεται χρήση συνθετικής υπερδειγματοληψίας παρατηρούμε ότι τα τυχαία δάση (Random Forests), ανταποκρίνονται και πάλι καλύτερα σε σχέση με τους υπόλοιπους αλγορίθμους αλλά με πολύ χαμηλότερη απόδοση 79,48%. Εδώ όμως, αρκετά καλά ανταποκρίνεται και ο αλγόριθμος Gaussian NB με ποσοστό 74,35%. Να σημειώσουμε εδώ ότι ο συγκεκριμένος αλγόριθμος, στην περίπτωση χρήσης συνθετικής υπερδειγματοληψίας ήταν ο χειρότερος σε απόδοση, ενώ στην περίπτωση που δεν γίνεται χρήση συνθετικής υπερδειγματοληψίας, η απόδοσή του βελτιώνεται, κατατάσσοντάς τον στην δεύτερη θέση. Ακολουθούν με χαμηλότερη απόδοση οι αλγόριθμοι Support Vector Machines (SVC) με ποσοστό 69,23 και Decision Tree με ποσοστό 66,66%.

Εδώ δίνεται και ο αντίστοιχος εκτελέσιμος κώδικας,

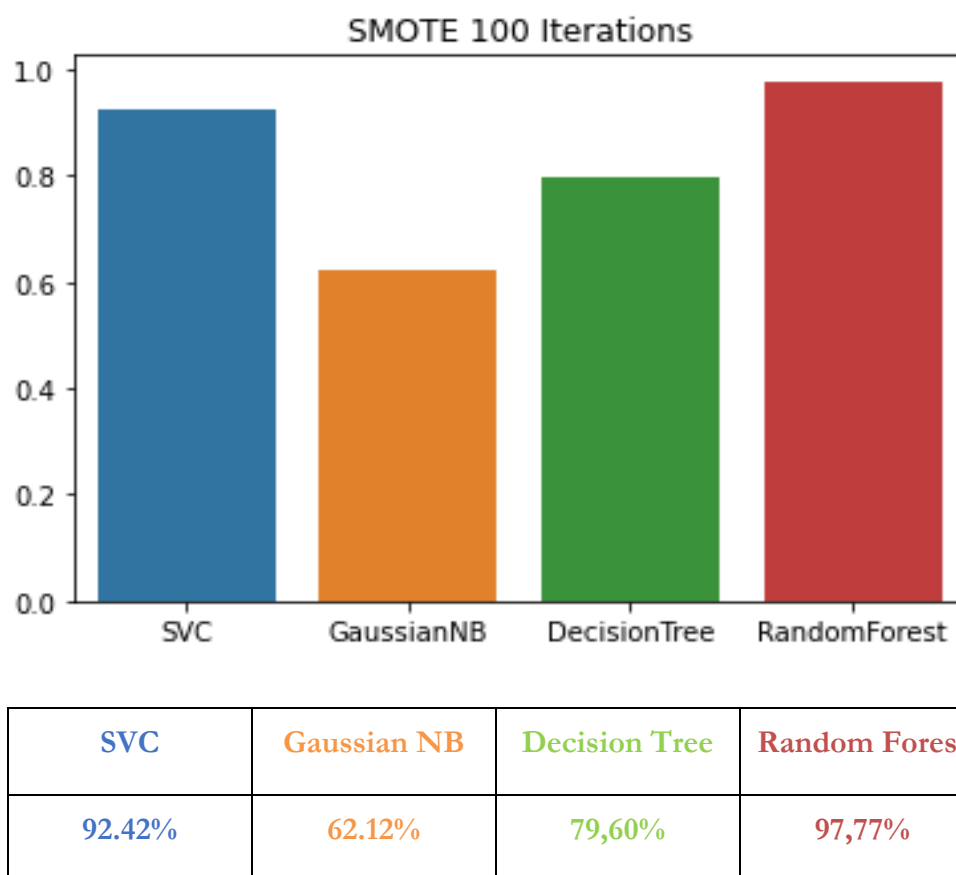
```
clf = SVC(kernel='poly',tol=3e-2,C=10, gamma="auto").fit(x_train, y_train)
clf1 = GaussianNB().fit(x_train, y_train)
clf2 = DecisionTreeClassifier().fit(x_train, y_train)
clf3 = RandomForestClassifier().fit(x_train, y_train)
score = clf.score(x_test, y_test)
score1 = clf1.score(x_test, y_test)
score2 = clf2.score(x_test, y_test)
score3 = clf3.score(x_test, y_test)
print(score)
print(score1)
print(score2)
print(score3)

y = [score, score1, score2, score3]
x = ["SVC", "GaussianNB", "DecisionTree", "RandomForest"]
# if the scores are changing it's rational since almost everytime we run this a different part of the data to validate.
sns.barplot(x=x, y=y).set_title('No SMOTE 1 Iteration')

0.6923076923076923
0.7435897435897436
0.6666666666666666
0.7948717948717948
```

Εικόνα 3-11. Εκτελέσιμος κώδικας μοντέλων (Χωρίς χρήση συνθετικής υπερδειγματοληψίας, 1 Επανάληψη).

Για να γίνουν τα παραπάνω αποτελέσματα ακόμα πιο αντιπροσωπευτικά εκτελέσαμε τον κώδικα μας για  $n=100$  επαναλήψεις και καταλήξαμε στα παρακάτω αποτελέσματα. Θα ξεκινήσουμε ομοίως την εκτέλεση με την εφαρμογή συνθετικής υπερδειγματοληψίας SMOTE.



Εικόνα 3-12. Barplot αποτελεσμάτων (Με χρήση συνθετικής υπερδειγματοληψίας, 100 Επαναλήψεις).

Και για  $n=100$  επαναλήψεις παρατηρούμε ότι τα τυχαία δάση (Random Forests), στην περίπτωση όπου εφαρμόζεται συνθετική υπερδειγματοληψία SMOTE ανταποκρίνονται πολύ καλά με απόδοση 97,77%. Ομοίως, πολύ καλά ανταποκρίνεται και ο αλγόριθμος Support Vector Machines (SVC) με απόδοση 92,42%, ενώ οι αλγόριθμοι Decision Tree με απόδοση 79,60% και Gaussian NB με απόδοση 62,12% δεν δίνουν ιδιαίτερα ικανοποιητικά αποτελέσματα.

Επίσης παρατίθεται και ο εκτελέσιμος κώδικας

```
from sklearn.svm import SVC
from sklearn.naive_bayes import GaussianNB
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import train_test_split
from sklearn.model_selection import cross_validate
from sklearn.ensemble import RandomForestClassifier

def avg(lst):
    return sum(lst) / len(lst)

svc = []
gb = []
dtc = []
rf = []

for i in range(100):
    clf = SVC(kernel='poly', tol=3e-2, C=10, gamma="auto").fit(train_x, train_y)
    clf1 = GaussianNB().fit(train_x, train_y)
    clf2 = DecisionTreeClassifier().fit(train_x, train_y)
    clf3 = RandomForestClassifier().fit(train_x, train_y)
    score = clf.score(test_x, test_y)
    score1 = clf1.score(test_x, test_y)
    score2 = clf2.score(test_x, test_y)
    score3 = clf3.score(test_x, test_y)

    svc.append(score)
    gb.append(score1)
    dtc.append(score2)
    rf.append(score3)

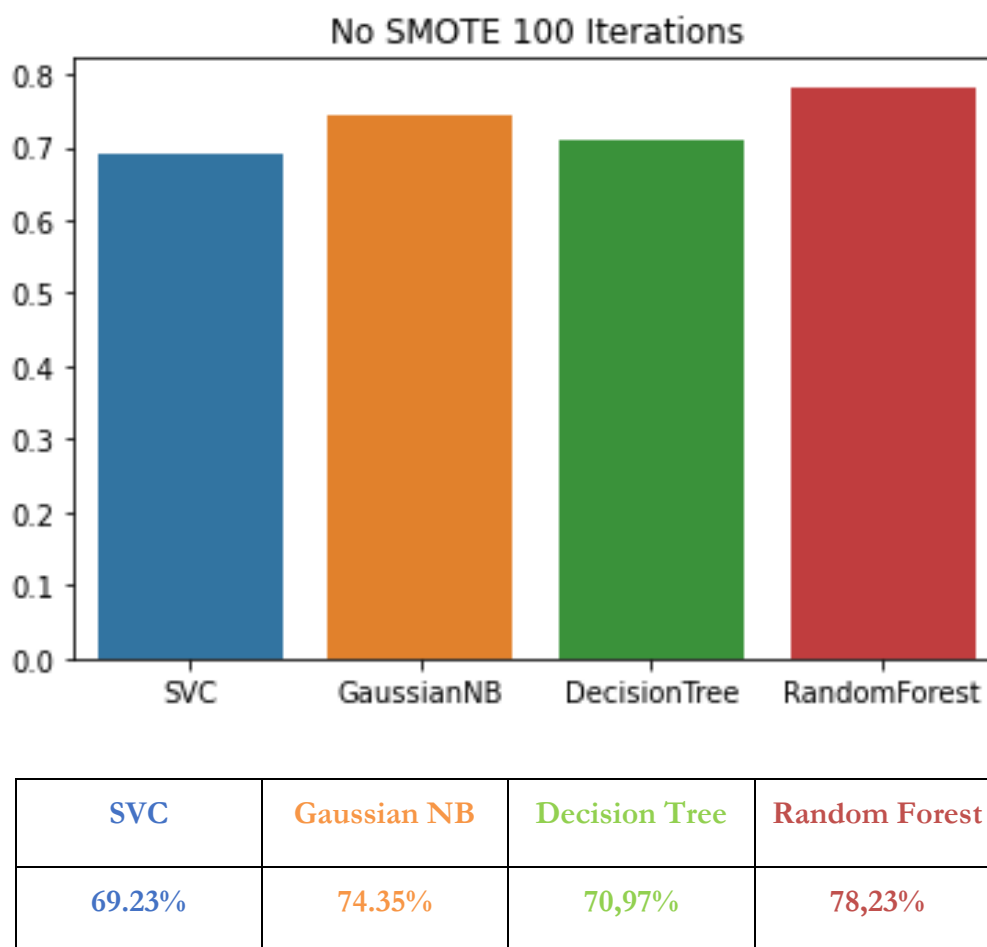
print(avg(svc))
print(avg(gb))
print(avg(dtc))
print(avg(rf))
y = [avg(svc), avg(gb), avg(dtc), avg(rf)]
x = ["SVC", "GaussianNB", "DecisionTree", "RandomForest"]
# if the scores are changing it's rational since almost everytime we run this a different part of the data to validate.
sns.barplot(x=x, y=y).set_title('SMOTE 100 Iterations')
```

0.9242424242424231  
0.6212121212121229  
0.7960606060606055  
0.9777272727272722

Εικόνα 3-13. Εκτελέσιμος κώδικας (Με χρήση συνθετικής υπερδειγματοληψίας, 100 Επανάληψεις).

Το ίδιο ακριβώς κάνουμε και για την περίπτωση χωρίς την χρήση συνθετικής υπερδειγματοληψίας SMOTE για να πάρουμε ξανά πιο αντιπροσωπευτικά αποτελέσματα.

Παρακάτω δίνεται το αντίστοιχο barplot μαζί με το accuracy table,



Εικόνα 3-14. Barplot αποτελεσμάτων (Χωρίς χρήση συνθετικής υπερδειγματοληψίας, 100 Επαναλήψεις).

Για  $n=100$  επαναλήψεις φαίνεται ξανά πόσο καθοριστική είναι η επιρροή της συνθετικής υπερδειγματοληψίας στην απόδοση των αλγορίθμων. Στην περίπτωση που δεν γίνεται χρήση συνθετικής υπερδειγματοληψίας παρατηρούμε ότι τα τυχαία δάση (Random Forests), ανταποκρίνονται και πάλι καλύτερα σε σχέση με τους υπόλοιπους αλγορίθμους αλλά με πολύ χαμηλότερη απόδοση 78,23%. Ομοίως, όπως και στην περίπτωση εκτέλεσης με  $n=1$  επανάληψη, αρκετά καλά ανταποκρίνεται ο αλγόριθμος Gaussian NB με ποσοστό 74,35%. Να σημειώσουμε ξανά ότι ο συγκεκριμένος αλγόριθμος, στην περίπτωση χρήσης συνθετικής υπερδειγματοληψίας ήταν ο χειρότερος σε απόδοση, ενώ στην περίπτωση που δεν γίνεται χρήση συνθετικής υπερδειγματοληψίας, η απόδοσή του βελτιώνεται, κατατάσσοντάς τον στην δεύτερη θέση. Ακολουθούν με χαμηλότερη απόδοση οι αλγόριθμοι Support Vector Machines (SVC) με ποσοστό 69,23 και Decision Tree με ποσοστό 70,97%, ο οποίος εδώ φαίνεται να βελτιώνει την απόδοσή του, καθώς με  $n=100$  επαναλήψεις, τα αποτελέσματά μας είναι πιο αντιπροσωπευτικά.

Επίσης δίνεται και αντίστοιχος ο εκτελέσιμος κώδικας.

```
from sklearn.svm import SVC
from sklearn.naive_bayes import GaussianNB
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import train_test_split
from sklearn.model_selection import cross_validate
from sklearn.ensemble import RandomForestClassifier

def avg(lst):
    return sum(lst) / len(lst)

svc = []
gb = []
dtc = []
rf = []

for i in range(100):
    clf = SVC(kernel='poly',tol=3e-2,C=10, gamma="auto").fit(x_train, y_train)
    clf1 = GaussianNB().fit(x_train, y_train)
    clf2 = DecisionTreeClassifier().fit(x_train, y_train)
    clf3 = RandomForestClassifier().fit(x_train, y_train)
    score = clf.score(x_test, y_test)
    score1 = clf1.score(x_test, y_test)
    score2 = clf2.score(x_test, y_test)
    score3 = clf3.score(x_test, y_test)

    svc.append(score)
    gb.append(score1)
    dtc.append(score2)
    rf.append(score3)

print(avg(svc))
print(avg(gb))
print(avg(dtc))
print(avg(rf))

y = [avg(svc), avg(gb), avg(dtc), avg(rf)]
x = ["SVC", "GaussianNB", "DecisionTree", "RandomForest"]
# if the scores are changing it's rational since almost everytime we run this a different part of the data to validate.
sns.barplot(x=x, y=y).set_title('No SMOTE 100 Iterations')
```

0.692307692307693  
0.7435897435897434  
0.7097435897435896  
0.7823076923076921

Εικόνα 3-15. Εκτελέσιμος κώδικας (Χωρίς χρήση συνθετικής υπερδειγματοληψίας, 100 Επανάληψεις).

## 4 ΑΠΟΤΕΛΕΣΜΑΤΑ

### 4.1 Συζήτηση Αποτελεσμάτων

Στο κεφάλαιο 3 παρουσιάστηκε η ανάλυση και λύση του προβλήματός μας, ξεκινώντας από την διαδικασία προ επεξεργασίας των λεκτικών δεδομένων, συνεχίζοντας με την διαδικασία εκμάθησης των μοντέλων μας και καταλήγοντας στην εξαγωγή των αποτελεσμάτων.

Πιο συγκεκριμένα, εφαρμόστηκαν οι μηχανισμοί επεξεργασίας φυσικού λόγου και οι αλγόριθμοι μηχανικής μάθησης που αναλύθηκαν στο κεφάλαιο 2, και συγκρίθηκαν μεταξύ τους ως προς την αποτελεσματικότητά τους. Συμπληρωματικά αποδείχτηκε η υπεροχή της τεχνικής Bag of Words έναντι της τεχνικής TFIDF και η συμβολή της τεχνικής της συνθετικής δειγματοληψίας στην εξαγωγή βελτιστοποιημένων αποτελεσμάτων. Ο στόχος όλων των παραπάνω είναι η αναγνώριση λεκτικών προτύπων από αναρτήσεις ιστολογίου (blog posts) και φόρουμ του σκοτεινού διαδικτύου (dark web), ώστε να βρεθεί η συσχέτιση των λέξεων ως προς το αν αυτά απευθύνονται σε τρομοκρατική στρατολόγηση ή όχι.

Στο κεφάλαιο αυτό θα γίνει η αξιολόγηση των αποτελεσμάτων και αναφορά στους τρόπους επέκτασης της παρούσας εργασίας και σχετικές εφαρμογές.

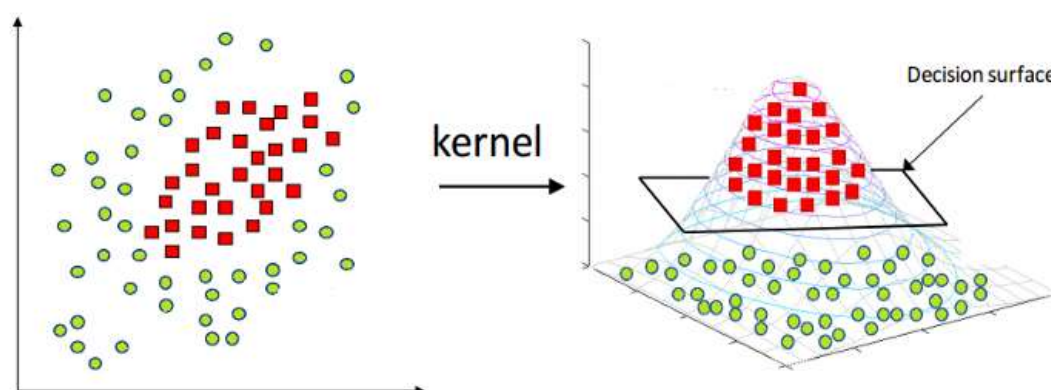
Μετά την παρουσίαση των αποτελεσμάτων όπου εδώ φαίνονται συγκεντρωμένα, μπορούν να γίνουν κάποια σχόλια.

	SVC	Gaussian NB	Decision Trees	Random Forests
Με SMOTE, 1 επανάληψη	92.42%	62.12%	77.27%	98.48%
Με SMOTE, 100 επαναλήψεις	92.42%	62.12%	79,60%	97,77%
Χωρίς SMOTE, 1 επανάληψη	69.23%	74.35%	66.66%	79.48%
Χωρίς SMOTE, 100 επαναλήψεις	69.23%	74.35%	70,97%	78,23%

Εικόνα 4-1.Συγκεντρωτικός πίνακας αποτελεσμάτων.

Είναι γνωστό ότι τα τυχαία δάση (Random Forests), ειδικά στην περίπτωση όπου εφαρμόζεται συνθετική υπερδειγματοληψία SMOTE ανταποκρίνονται πολύ καλά σε προβλήματα τα οποία έχουν να κάνουν με λεκτικά δεδομένα, αυτό άλλωστε αποδεικνύεται και από τα αποτελέσματά μας. Πολύ καλά επίσης ανταποκρίνεται και ο αλγόριθμος Support Vector Machines (SVM). Σε αυτό συνεργεί το “Kernel Trick” το οποίο δίνει την δυνατότητα στον αλγόριθμο να μετατρέπει την προσέγγιση του από γραμμική σε μη γραμμική. Ο μηχανισμός αυτός, προσδίδει μεγάλο πλεονέκτημα σε αυτό το μοντέλο καθώς χωρίς αυτόν, το μοντέλο θα ήταν γραμμικό και όπως φαίνεται από τα αποτελέσματα του “Gaussian NB”, οι γραμμικοί αλγόριθμοι δεν ανταποκρίνονται αρκετά καλά στο πρόβλημα μας.

Η αναφορά του όρου Kernel Trick στα μοντέλα διανυσματικών μηχανών υποστήριξης SVM υπονοεί τη γεφύρωση της γραμμικότητας και της μη γραμμικότητας. Στον πραγματικό κόσμο, σχεδόν όλα τα δεδομένα κατανέμονται τυχαία, γεγονός που καθιστά δύσκολο τον διαχωρισμό διαφορετικών τάξεων γραμμικά. Μια σιέψη για αυτήν τη διαδικασία μετατροπής δεδομένων είναι η χαρτογράφηση όλων των σημείων δεδομένων σε μια υψηλότερη διάσταση, να βρεθεί το όριο διαχωρισμού και να γίνει η ταξινόμηση.



Εικόνα 4-2. Kernel Trick - η χαρτογράφηση όλων των σημείων δεδομένων σε υψηλότερη διάσταση.

Ωστόσο, όταν υπάρχουν όλο και περισσότερες διαστάσεις, οι υπολογισμοί εντός αυτού του χώρου γίνονται όλο και πιο υπολογιστικά “ακριβοί”. Αυτό που κάνει το Kernel Trick για εμάς είναι να προσφέρει έναν πιο αποτελεσματικό και λιγότερο “ακριβό” υπολογιστικά τρόπο για τη μετατροπή των δεδομένων σε υψηλότερες διαστάσεις.

Στην περίπτωση όπου δεν εφαρμόζεται η συνθετική υπερδειγματοληψία (SMOTE), παρατηρούμε ότι κανένας αλγόριθμος δεν εξάγει ικανοποιητικά αποτελέσματα καθώς τα δεδομένα είναι λίγα και η διαστατικότητα μικρή. Η ανισορροπία στα δεδομένα (Data Imbalance) κάτι που αναλύθηκε στο προηγούμενο κεφάλαιο, δεν εξάγει ικανοποιητικά αποτελέσματα καθώς δεν λαμβάνονται βελτιστοποιημένα αποτελέσματα για την τάξη που είναι ασύμμετρη και το μοντέλο δεν λαμβάνει ποτέ επαρκή εικόνα για την υποκείμενη



κατηγορία. Επίσης δημιουργούν πρόβλημα στην διαδικασία της δημιουργίας ενός testing set καθώς στην περίπτωση μας, λόγω των λιγοστών δειγμάτων που έχουμε για την ασύμμετρη κλάση, κατά πάσα πιθανότητα θα έχουμε και λίγα δείγματα για αυτήν στο testing set.

Στην περίπτωση όπου δεν εφαρμόζεται η συνθετική υπερδειγματοληψία (SMOTE), παρατηρείται ότι ο αλγόριθμος Random Forest είναι ξανά ο πιο αποτελεσματικός, ενώ συγκριτικά καλά ανταποκρίνεται και ο αλγόριθμος Gaussian NB. Ειδικότερα ο αλγόριθμος Gaussian NB στην περίπτωση μη εφαρμογής συνθετικής υπερδειγματοληψίας (SMOTE) ανταποκρίνεται καλύτερα από ότι στην περίπτωση εφαρμογής της. Ο αλγόριθμος Decision Tree φαίνεται ότι ωφελείται από την εφαρμογή συνθετικής υπερδειγματοληψίας (SMOTE), αλλά δυστυχώς, τα αποτελέσματα σε καμία περίπτωση δεν είναι ικανοποιητικά.

## 4.2 Τρόποι Επέκτασης.

Παρακάτω παρατίθενται κάποιοι τρόποι επέκτασης της παρούσας εργασίας.

1. **Περισσότερα Δεδομένα:** Στον τομέα της Μηχανικής Μάθησης είναι βασική αρχή πως όσα περισσότερα και καλύτερης ποιότητας δεδομένα υπάρχουν, τόσο καλύτερα και πιο αντιπροσωπευτικά αποτελέσματα μπορούμε να πετύχουμε.
2. **Βαθιά Μάθηση (Δεδομένου του 1.):** Στο τομέα της βαθιάς μάθησης υπάρχουν αρκετά μοντέλα τα οποία είναι πολύ καλύτερα στο να αναλύουν λεκτικά δεδομένα, όμως αυτά τα μοντέλα προϋποθέτουν την ύπαρξη πολλών δειγμάτων και για αυτό στην παρούσα διπλωματική δεν χρησιμοποιήθηκαν τέτοιες τεχνικές αρχιτεκτονικής μοντέλων.  
Στην περίπτωση λοιπόν που υπήρχαν περισσότερα δεδομένα, ένα τέτοιο μοντέλο θα μπορούσε να είναι κάποιο από την οικογένεια των LSTM neural networks τα οποία είναι πολύ γνωστά για την ικανότητα τους να αναλύουν ακολουθίες κειμένου.
3. **Προσθήκη λογισμικού μετάφρασης:** Πολλά από τα δεδομένα που μπορούμε να εισάγουμε σε μοντέλο μηχανικής μάθησης είναι γραμμένα σε άλλες γλώσσες πέρα των Αγγλικών. Θα μπορούσε λοιπόν το μοντέλο το οποίο εκπαιδεύτηκε στα πλαίσια αυτής της εργασίας να επεκταθεί, προσθέτοντας ένα επιπλέον επίπεδο (layer) όπου μέσα σε αυτό, τα μηνύματα να μεταφράζονται από κάποιο κατάλληλο λογισμικό, για παράδειγμα από το API (Application programming interface) μετάφρασης της Google το οποίο στις μέρες μας είναι πολύ πιο αποτελεσματικό σε σχέση με τα προηγούμενα χρόνια.
4. **Προσθήκη λογισμικού μετατροπής ομιλίας σε κείμενο:** Εάν έχουμε στην διάθεσή μας δεδομένα τα οποία βρίσκονται σε μορφή ηχητικού αρχείου, θα μπορούσε το μοντέλο το οποίο εκπαιδεύτηκε στα πλαίσια αυτής της εργασίας να επεκταθεί, προσθέτοντας ένα επιπλέον επίπεδο (layer) όπου μέσα

σε αυτό, τα μηνύματα να μετατατρέπονται από κάποιο κατάλληλο λογισμικό σε αρχεία κειμένου, ώστε να είναι συμβατά για χρήση στο μοντέλο μας.

Πέρα από τα παραπάνω και μετά από εκτεταμένη έρευνα, υπάρχει η πεποίθηση πως καταφέραμε να φτάσουμε στα καλύτερα δυνατά αποτελέσματα, δεδομένου του διαθέσιμου δείγματος.

### 4.3 Εφαρμογές

Συμπερασματικά φαίνεται ότι ένα εκπαιδευμένο μοντέλο μηχανικής μάθησης που είναι ικανό να αναγνωρίζει αν ένα κομμάτι κειμένου είναι στοχευμένο στην στρατολόγηση ατόμων για τρομοκρατικές επιθέσεις, θα μπορούσε εύκολα να χρησιμοποιηθεί για την αντιμετώπιση και την πρόληψη αυτών, καθώς ο όγκος των μηνυμάτων που βρίσκονται και προστίθενται στα forums του dark web είναι τόσο μεγάλος, που καθίσταται σχεδόν αδύνατο να υπάρχει ανθρώπινο δυναμικό που να βρίσκεται μόνιμα εκεί ώστε να τα διαβάζει και να τα αποκωδικοποιεί.

Το πεδίο εφαρμογών είναι πραγματικά ανεξάντλητο, αλλά επιγραμματικά θα μπορούσαν να αναφερθούν κάποιες που έχουν κάποια συνάφεια με την παρούσα εργασία, όπως:

- η αντιμετώπιση και πρόληψη της παράνομης πορνογραφίας
- η αντιμετώπιση και πρόληψη παράνομων οικονομικών συναλλαγών
- η αντιμετώπιση και πρόληψη διακίνησης ναρκωτικών
- η αντιμετώπιση και πρόληψη παράνομης εμπορίας όπλων,
- η αντιμετώπιση και πρόληψη παραγωγής και διακίνησης πλαστών νομισμάτων
- η αντιμετώπιση και πρόληψη παράνομης διακίνησης – εμπορίας ανθρώπων
- η αντιμετώπιση και πρόληψη του λαθρεμπορίου
- και πολλές άλλες.

## ΒΙΒΛΙΟΓΡΑΦΙΑ

- Ada Lovelace. Wikipedia. [https://en.wikipedia.org/wiki/Ada\\_Lovelace](https://en.wikipedia.org/wiki/Ada_Lovelace)
- Rong Jin, Zhi-Hua Zhou (2010). Understanding bag-of-words model: A statistical framework.
- Shahzad Qaiser, Ramsha Ali (2018). Text Mining: Use of TF-IDF to Examine the Relevance of Words to Documents.
- Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, W. Philip Kegelmeyer (2002). SMOTE: Synthetic Minority Over-sampling Technique.
- Simon Tong, Daphne Koller (2001). Support Vector Machine Active Learning with Applications to Text Classification.
- Leo Breiman (2001). RANDOM FORESTS.
- Quinlan, J. R. (1986). Induction of Decision Trees. Mach. Learn.
- Vapnik, V. N.; Chervonenkis, A. Ya. (1971). "On the Uniform Convergence of Relative Frequencies of Events to Their Probabilities". Theory of Probability & Its Applications.
- Cortes, Corinna; Vapnik, Vladimir (1995-09-01). "Support-vector networks". Machine Learning.
- Ιωάννης Καβελάρης. Αθήνα, (Μάρτιος 2017) - Πρακτικές χρήσης μέσων κοινωνικής δικτύωσης σε ανθρωπογενείς καταστροφές από δημόσιους φορείς. Περιπτώσεις Βοστώνης, Παρισιού και Νίκαιας, Βρυξελλών, Τουρκίας και ελληνική διάσταση του θέματος.
- Kalev Leetaru. (May 2019). Countering Online Extremism Is Too Important To Leave To Facebook. Forbes magazine.
- Γνωμοδότηση της Ευρωπαϊκής Οικονομικής και Κοινωνικής Επιτροπής με θέμα «Πρόταση κανονισμού του Ευρωπαϊκού Κοινοβουλίου και του Συμβουλίου σχετικά με την πρόληψη της διάδοσης τρομοκρατικού περιεχομένου στο διαδίκτυο» [COM(2018) 640 final — 2018-0331 (COD)](2019/C 110/13)
- Chris Albon (2018). Machine Learning with Python Cookbook. O'Reilly Media, Inc.
- Jason Brownlee. Master Machine Learning Algorithms Ebook.
- J. Huang; J. Lu; C.X. Ling (2003). Comparing naive Bayes, decision trees, and SVM with AUC and accuracy. Third IEEE International Conference on Data Mining



## ΠΑΡΑΡΤΗΜΑ Α: ΚΩΔΙΚΑΣ

Counter Terrorism on Forums

Training part

in this part i'll write the training code so that you can run it.

```
import numpy as np
import pandas as pd
import nltk
#uncomment these two lines above the first time you run it and download all the nltk
packages through the UI that will open
#then recomment them.
```

```
data = pd.read_excel('Ansar1_Original.xlsx')[:193]
data.head()
```

	Unnamed: 0	Unnamed: 1	Unnamed: 2	JacobRaw	Unnamed: 4	Unnamed: 5
0	AgreeID	MessageID	ThreadID	Recruitment	Translation	Message
1	1	31599	11251	0	NaN	{Source: Digital Chicago, Inc} North Side resi...
2	2	117	98	1	NaN	In the name of God the Merciful Praise be to A...
3	3	1606	520	0	NaN	Did Mansoor join the emerat?I heard he is stil...
4	4	5644	188	0	NaN	I dont think \juba\" is one person. from iai s...

```
from nltk.tokenize import word_tokenize
import string
from nltk.corpus import stopwords
from nltk.stem import WordNetLemmatizer
from nltk.stem.porter import PorterStemmer

text = []
for message in data['Unnamed: 5'].values[1:]:
    text.append([message])

def text_proc(data):

    for message in range(len(data)):
```

```
data[message] = word_tokenize(data[message][0])
# convert to lower case
data[message] = [w.lower() for w in data[message]]
# remove punctuation from each word
table = str.maketrans('', '', string.punctuation)
data[message] = [w.translate(table) for w in data[message]]
# remove remaining tokens that are not alphabetic
words_nl = [word for word in data[message] if word.isalpha()]
# filter out stop words whitespaces etc.
stop_words = set(stopwords.words('english'))
data[message] = [w for w in words_nl if not w in stop_words]
lemmatizer = WordNetLemmatizer() # use the porter stemmer to stem all the words
data[message] = [lemmatizer.lemmatize(word) for word in data[message]]
return data
```

```
text = text_proc(text)
print(text[0:2]) # print two samples 0 to 2
```

```
labels = data['JacobRaw'].values.tolist()[1:]
```

```
labels = np.array(labels)
```

```
[['source', 'digital', 'chicago', 'inc', 'north', 'side', 'resident', 'uzair', 'ali',
'hashmi', 'informed', 'fbi', 'last', 'month', 'terrorist', 'trying', 'recruit', 'one',
'gave', 'letter', 'stating', 'job', 'carry', 'mission', 'giving', 'nonbeliever',
'deserve', 'hashmi', 'said', 'letter', 'also', 'asked', 'familiar', 'downtown',
'chicago', 'stated', 'calling', 'jihad', 'brother', 'key', 'role', 'operation',
'investigation', 'found', 'hashmi', 'made', 'story', 'fabricated', 'letter',
'authority', 'said', 'wednesday', 'indicted', 'federal', 'grand', 'jury', 'three',
'count', 'making', 'false', 'statement', 'fbi', 'according', 'indictment', 'hashmi',
'contacted', 'fbi', 'july', 'saying', 'someone', 'approached', 'previous', 'day',
'asked', 'proficient', 'firearm', 'suggested', 'join', 'god', 'military', 'person',
'hashmi', 'claimed', 'approached', 'actually', 'made', 'statement', 'indictment',
'said', 'take', 'seriously', 'allegation', 'terrorism', 'activity', 'aggressively',
'investigate', 'every', 'lead', 'said', 'robert', 'grant', 'special', 'agentincharge',
'chicago', 'fbi', 'want', 'encourage', 'people', 'report', 'genuinely', 'suspicious',
```

```
'activity', 'also', 'seek', 'prosecute', 'anyone', 'deliberately', 'provides', 'false',  
'information', 'diverts', 'agent', 'resource', 'important', 'matter', 'convicted',  
'hashmi', 'could', 'get', 'year', 'prison', 'fine', 'scheduled', 'arraigned',  
'wednesday', 'u', 'district', 'court'], ['name', 'god', 'merciful', 'praise', 'allah',  
'enough', 'enough', 'prayer', 'peace', 'upon', 'prophet', 'chosen', 'ansar',  
'almujahideen', 'progress', 'incitement', 'u', 'people', 'sunni', 'community',  
'lebanon', 'join', 'fatah', 'alislam', 'supporting', 'need', 'thanks', 'god',  
'almighty', 'produce', 'version', 'voiceprovocation', 'incite', 'young', 'sunni',  
'brave', 'lebanon', 'god', 'brother', 'fatah', 'alislam', 'defend', 'support',  
'despite', 'tkhalakm', 'epic', 'nahr', 'albared', 'taidoa', 'sin', 'may', 'god',  
'forgive', 'u', 'every', 'nation', 'unification', 'version', 'new', 'voice', 'occasion',  
'eid', 'sunni', 'lebanon', 'download', 'file', 'size', 'high', 'quality', 'mb',  
'forget', 'mujahideen', 'mesh', 'دعائكم', 'fit', 'ansar', 'almujahideen']]
```

```
join_text = []  
for message in range(len(text)):  
    join_text.append(' '.join(text[message])) # join all the text to feed it into the  
bag of words.
```

```
tf_text = join_text  
import collections, re  
#create the bagofwords  
bagsofwords = [collections.Counter(re.findall(r'\w+', txt)) for txt in join_text]  
sumbags = sum(bagsofwords, collections.Counter())
```

```
# transform all the words from our features matrix into frequency numerics
```

```
for message in text:  
    for i in range(len(message)):  
        message[i] = sumbags[str(message[i])]
```

```
print(text[0:2]) # check if the transform was successful
```

```
[[55, 2, 4, 1, 43, 19, 10, 1, 17, 6, 6, 33, 50, 69, 73, 25, 70, 159, 16, 13, 1, 10, 18,  
8, 12, 2, 4, 6, 502, 13, 166, 26, 4, 3, 4, 11, 9, 132, 83, 18, 10, 53, 23, 18, 6, 32,  
10, 1, 13, 64, 502, 18, 2, 19, 3, 2, 52, 9, 16, 6, 29, 33, 66, 13, 6, 2, 33, 17, 34, 18,  
5, 4, 54, 26, 2, 3, 5, 131, 30, 91, 30, 6, 17, 5, 7, 32, 29, 13, 502, 50, 5, 3, 46, 17,  
2, 2, 38, 8, 502, 5, 10, 25, 1, 4, 33, 44, 7, 174, 49, 1, 4, 17, 166, 17, 1, 16, 2, 2,
```

```
6, 24, 1, 17, 10, 14, 8, 10, 6, 41, 42, 127, 30, 1, 3, 1, 18, 287, 25, 42], [23, 30, 5,
13, 130, 13, 13, 10, 25, 38, 7, 3, 4, 2, 6, 1, 287, 174, 13, 37, 6, 131, 2, 3, 14, 33,
3, 30, 6, 5, 5, 1, 1, 55, 13, 1, 6, 30, 83, 2, 3, 11, 65, 22, 1, 1, 1, 1, 1, 3, 107, 30,
2, 287, 38, 31, 1, 5, 77, 5, 3, 1, 13, 6, 3, 1, 5, 15, 2, 7, 2, 77, 1, 1, 2, 4, 2]]
```

```
from sklearn.feature_extraction.text import TfidfVectorizer
```

```
tfidf = TfidfVectorizer()
```

```
tf_text = tfidf.fit_transform(tf_text)
```

```
text = np.array(text)
```

```
seq_len = 1100
```

```
features = np.zeros((len(text), seq_len), dtype=int)
```

```
for i, row in enumerate(text):
```

```
    features[i, -len(row):] = np.array(row)[:seq_len]
```

```
import seaborn as sns
```

```
%matplotlib inline
```

```
sns.countplot(labels)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x1b4c82e5dc0>
```

We're gonna use smote to fix the oversampling problem that we clearly have.

```
from imblearn.over_sampling import SMOTE
```

```
from sklearn.model_selection import train_test_split
```

```
sm = SMOTE(random_state=33)
```

```
X_train_new, y_train_new = sm.fit_sample(features, labels.ravel())
```

```
train_x, test_x, train_y, test_y = train_test_split(X_train_new, y_train_new,
test_size=0.2, random_state=100)
```

```
x_train, x_test, y_train, y_test = train_test_split(features, labels, test_size=0.2,
random_state=100)
```

```
sns.countplot(y_train_new)
```



```
x_train_tfidf, x_test_tfidf, y_train_tfidf, y_test_tfidf = train_test_split(tf_text, labels,  
test_size=0.2, random_state=100)
```

```
#helper and plotting functions are gonna be here  
import matplotlib.pyplot as plt  
from sklearn.model_selection import learning_curve  
def plot_confusion_matrix(y_true, y_pred, classes,  
                           normalize=False,  
                           title=None,  
                           cmap=plt.cm.Blues):  
    """  
    This function prints and plots the confusion matrix.  
    Normalization can be applied by setting `normalize=True`.  
    """  
    if not title:  
        if normalize:  
            title = 'Normalized confusion matrix'  
        else:  
            title = 'Confusion matrix, without normalization'  
  
    # Compute confusion matrix  
    cm = confusion_matrix(y_true, y_pred)  
    # Only use the labels that appear in the data  
    classes = classes[unique_labels(y_true, y_pred)]  
    if normalize:  
        cm = cm.astype('float') / cm.sum(axis=1)[:, np.newaxis]  
        print("Normalized confusion matrix")  
    else:  
        print('Confusion matrix, without normalization')  
  
    print(cm)  
  
    fig, ax = plt.subplots()  
    im = ax.imshow(cm, interpolation='nearest', cmap=cmap)  
    ax.figure.colorbar(im, ax=ax)  
    # We want to show all ticks...  
    ax.set(xticks=np.arange(cm.shape[1]),  
           yticks=np.arange(cm.shape[0]),
```

```

        # ... and label them with the respective list entries
        xticklabels=classes, yticklabels=classes,
        title=title,
        ylabel='True label',
        xlabel='Predicted label')

# Rotate the tick labels and set their alignment.
plt.setp(ax.get_xticklabels(), rotation=45, ha="right",
         rotation_mode="anchor")

# Loop over data dimensions and create text annotations.
fmt = '.2f' if normalize else 'd'
thresh = cm.max() / 2.
for i in range(cm.shape[0]):
    for j in range(cm.shape[1]):
        ax.text(j, i, format(cm[i, j], fmt),
                ha="center", va="center",
                color="white" if cm[i, j] > thresh else "black")
fig.tight_layout()
return ax

def plot_learning_curve(estimator, title, X, y, ylim=None, cv=None,
                        n_jobs=None, train_sizes=np.linspace(.1, 1.0, 5)):
    plt.figure()
    plt.title(title)
    if ylim is not None:
        plt.ylim(*ylim)
    plt.xlabel("Training examples")
    plt.ylabel("Score")
    train_sizes, train_scores, test_scores = learning_curve(
        estimator, X, y, cv=cv, n_jobs=n_jobs, train_sizes=train_sizes)
    train_scores_mean = np.mean(train_scores, axis=1)
    train_scores_std = np.std(train_scores, axis=1)
    test_scores_mean = np.mean(test_scores, axis=1)
    test_scores_std = np.std(test_scores, axis=1)
    plt.grid()

    plt.fill_between(train_sizes, train_scores_mean - train_scores_std,

```

```
        train_scores_mean + train_scores_std, alpha=0.1,
        color="r")
plt.fill_between(train_sizes, test_scores_mean - test_scores_std,
                 test_scores_mean + test_scores_std, alpha=0.1, color="g")
plt.plot(train_sizes, train_scores_mean, 'o-', color="r",
         label="Training score")
plt.plot(train_sizes, test_scores_mean, 'o-', color="g",
         label="Cross-validation score")

plt.legend(loc="best")
return plt

from sklearn.svm import SVC
from sklearn.naive_bayes import GaussianNB
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import train_test_split
from sklearn.model_selection import cross_validate
from sklearn.ensemble import RandomForestClassifier

clf = SVC(kernel='poly',tol=3e-2,C=10, gamma="auto").fit(train_x, train_y)
clf1 = GaussianNB().fit(train_x, train_y)
clf2 = DecisionTreeClassifier().fit(train_x, train_y)
clf3 = RandomForestClassifier().fit(train_x, train_y)
clf4 = RandomForestClassifier().fit(x_train_tfidf, y_train_tfidf)
score = clf.score(test_x, test_y)
score1 = clf1.score(test_x, test_y)
score2 = clf2.score(test_x, test_y)
score3 = clf3.score(test_x, test_y)
score4 = clf4.score(x_test_tfidf, y_test_tfidf)

print(score)
print(score1)
print(score2)
print(score3)
print("tfidf score with Random Forest " + str(score4))

y = [score, score1, score2, score3]
x = ["SVC", "GaussianNB", "DecisionTree", "RandomForest"]
```

```
# if the scores are changing it's rational since almost everytime we run this a  
different part of the data to validate.
```

```
sns.barplot(x=x, y=y).set_title('SMOTE 1 Iteration')
```

```
0.9242424242424242
```

```
0.6212121212121212
```

```
0.7727272727272727
```

```
0.9848484848484849
```

```
tfidf score with Random Forest 0.8205128205128205
```

```
Text(0.5, 1.0, 'SMOTE 1 Iteration')
```

```
clf = SVC(kernel='poly',tol=3e-2,C=10, gamma="auto").fit(x_train, y_train)
```

```
clf1 = GaussianNB().fit(x_train, y_train)
```

```
clf2 = DecisionTreeClassifier().fit(x_train, y_train)
```

```
clf3 = RandomForestClassifier().fit(x_train, y_train)
```

```
score = clf.score(x_test, y_test)
```

```
score1 = clf1.score(x_test, y_test)
```

```
score2 = clf2.score(x_test, y_test)
```

```
score3 = clf3.score(x_test, y_test)
```

```
print(score)
```

```
print(score1)
```

```
print(score2)
```

```
print(score3)
```

```
y = [score, score1, score2, score3]
```

```
x = ["SVC", "GaussianNB", "DecisionTree", "RandomForest"]
```

```
# if the scores are changing it's rational since almost everytime we run this a  
different part of the data to validate.
```

```
sns.barplot(x=x, y=y).set_title('No SMOTE 1 Iteration')
```

```
0.6923076923076923
```

```
0.7435897435897436
```

```
0.6666666666666666
```

```
0.8461538461538461
```

```
Text(0.5, 1.0, 'No SMOTE 1 Iteration')

from sklearn.svm import SVC
from sklearn.naive_bayes import GaussianNB
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import train_test_split
from sklearn.model_selection import cross_validate
from sklearn.ensemble import RandomForestClassifier

def avg(lst):
    return sum(lst) / len(lst)

svc = []
gb = []
dtc = []
rf = []

for i in range(100):
    clf = SVC(kernel='poly',tol=3e-2,C=10, gamma="auto").fit(train_x, train_y)
    clf1 = GaussianNB().fit(train_x, train_y)
    clf2 = DecisionTreeClassifier().fit(train_x, train_y)
    clf3 = RandomForestClassifier().fit(train_x, train_y)
    score = clf.score(test_x, test_y)
    score1 = clf1.score(test_x, test_y)
    score2 = clf2.score(test_x, test_y)
    score3 = clf3.score(test_x, test_y)

    svc.append(score)
    gb.append(score1)
    dtc.append(score2)
    rf.append(score3)

print(avg(svc))
print(avg(gb))
print(avg(dtc))
print(avg(rf))
y = [avg(svc), avg(gb), avg(dtc), avg(rf)]
x = ["SVC", "GaussianNB", "DecisionTree", "RandomForest"]
```

```
# if the scores are changing it's rational since almost everytime we run this a  
different part of the data to validate.
```

```
sns.barplot(x=x, y=y).set_title('SMOTE 100 Iterations')
```

```
0.9242424242424231
```

```
0.6212121212121229
```

```
0.794545454545454
```

```
0.9790909090909087
```

```
Text(0.5, 1.0, 'SMOTE 100 Iterations')
```

```
from sklearn.svm import SVC
```

```
from sklearn.naive_bayes import GaussianNB
```

```
from sklearn.tree import DecisionTreeClassifier
```

```
from sklearn.model_selection import train_test_split
```

```
from sklearn.model_selection import cross_validate
```

```
from sklearn.ensemble import RandomForestClassifier
```

```
def avg(lst):
```

```
    return sum(lst) / len(lst)
```

```
svc = []
```

```
gb = []
```

```
dtc = []
```

```
rf = []
```

```
for i in range(100):
```

```
    clf = SVC(kernel='poly',tol=3e-2,C=10, gamma="auto").fit(x_train, y_train)
```

```
    clf1 = GaussianNB().fit(x_train, y_train)
```

```
    clf2 = DecisionTreeClassifier().fit(x_train, y_train)
```

```
    clf3 = RandomForestClassifier().fit(x_train, y_train)
```

```
    score = clf.score(x_test, y_test)
```

```
    score1 = clf1.score(x_test, y_test)
```

```
    score2 = clf2.score(x_test, y_test)
```

```
    score3 = clf3.score(x_test, y_test)
```

```
    svc.append(score)
```

```
    gb.append(score1)
```

```
dtc.append(score2)
rf.append(score3)

print(avg(svc))
print(avg(gb))
print(avg(dtc))
print(avg(rf))

y = [avg(svc), avg(gb), avg(dtc), avg(rf)]
x = ["SVC", "GaussianNB", "DecisionTree", "RandomForest"]
# if the scores are changing it's rational since almost everytime we run this a
different part of the data to validate.
sns.barplot(x=x, y=y).set_title('No SMOTE 100 Iterations')

0.692307692307693
0.7435897435897434
0.7123076923076916
0.7802564102564101

Text(0.5, 1.0, 'No SMOTE 100 Iterations')
```