

Πανεπιστήμιο Πατρών

Τμήμα Μηχανικών Η/Υ και Πληροφορικής

**Εξόρυξη Δεδομένων και Αλγόριθμοι Μάθησης**

**Υλοποιητικό Project 2025**

Κωνσταντίνος Αναστασόπουλος 1093320

Θεόφραστος Παξιμάδης 1093460

Table of Contents

[Δομή Προγράμματος 3](#_Toc199355905)

[Ερώτημα 1 4](#_Toc199355906)

[Περιγραφή Υλοποίησης 4](#_Toc199355907)

[Πληροφορίες Στηλών 4](#_Toc199355908)

[Συγκεντρωτικά Στατιστικά Μεγέθη 6](#_Toc199355909)

[Γραφικές Αναπαραστάσεις 7](#_Toc199355910)

[Συσχετίσεις 10](#_Toc199355911)

[Ερώτημα 2: Μείωση Συνόλου Δεδομένων 13](#_Toc199355912)

[Μείωση στηλών 13](#_Toc199355913)

[Δειγματοληψία 14](#_Toc199355914)

[Ερώτημα 3 16](#_Toc199355915)

[Περιγραφή Υλοποίησης 16](#_Toc199355916)

[Αξιολόγηση 17](#_Toc199355917)

[Sampled Data, Binary Classification, MLP 17](#_Toc199355918)

[Sampled Data, Binary Classification, SVM 17](#_Toc199355919)

[Sampled Data, Multiclass Classification, MLP 18](#_Toc199355920)

[Sampled Data, Multiclass Classification, SVM 18](#_Toc199355921)

[K-means Data, Binary Classification, MLP 19](#_Toc199355922)

[K-means Data, Binary Classification, SVM 19](#_Toc199355923)

[K-means Data, Multiclass Classification, MLP 20](#_Toc199355924)

[K-means Data, Multiclass Classification, SVM 20](#_Toc199355925)

[Birch Data, Binary Classification, MLP 21](#_Toc199355926)

[Birch Data, Binary Classification, SVM 21](#_Toc199355927)

[Birch Data, Multiclass Classification, MLP 22](#_Toc199355928)

[Birch Data, Multiclass Classification, SVM 22](#_Toc199355929)

[Σχολιασμός Αποτελεσμάτων 23](#_Toc199355930)

# Δομή Προγράμματος

Ο κώδικας για την υλοποίηση των ερωτημάτων γράφτηκε σε Python στο περιβάλλον ανάπτυξης Visual Studio Code. Οι βιβλιοθήκες που χρησιμοποιήθηκαν ήταν οι: pandas, matplotlib, io, pathlib, time, sklearn και numpy.

Η εγκατάστασή τους γίνεται πληκτρολογώντας στο terminal, στο κατάλληλο folder:

pip install pandas

pip install numpy

pip install matplotlib

pip install scikit-learn

Για να τρέξει το πρόγραμμα απαιτείται το αρχείο δεδομένων να ονομάζεται “data.csv” και να βρίσκεται σε folder με όνομα “data”, ο οποίος πρέπει να βρίσκεται στον ίδιο folder πατέρα με τον folder των αρχείων κώδικα. Τα αρχεία κώδικα πρέπει να βρίσκονται στον folder “src”. Τέλος ο folder “documentation” περιέχει την pdf αναφορά. Όλα τα αρχεία και folders έχουν σταλεί σε αυτή τη δομή στο zip, με εξαίρεση το dataset. Η δομή του αρχείου είναι η ακόλουθη.

deliverables/

├── src/

│ ├── question1.py

│ ├── question2.py

│ └── question3.py

├── data/

│ └── data.csv

└── documentation/

└── report.pdf

# Ερώτημα 1

## Περιγραφή Υλοποίησης

### Πληροφορίες Στηλών

Στο παρόν ερώτημα έχει χρησιμοποιηθεί για τα υποερωτήματα, κατά κύριο λόγο η βιβλιοθήκη pandas. Το πρώτο ζητούμενο που αφορά την εύρεση πληροφοριών σχετικά με τις στήλες του dataset υλοποιείται μέσω της συνάρτησης info() της προαναφερθείσας βιβλιοθήκης. Το πρόγραμμα κάνει export ως csv τα αποτελέσματα στον folder “data” στο αρχείο “info.csv”. Το περιεχόμενο του αρχείου φαίνεται παρακάτω.

|  |  |
| --- | --- |
| Info | |
| <class 'pandas.core.frame.DataFrame'> | |
| RangeIndex: 8656767 entries, 0 to 8656766 | |
| Data columns (total 86 columns): | |
| # Column Dtype | |
| --- ------ ----- | |
| 0 Flow ID object | |
| 1 Src IP object | |
| 2 Src Port float64 | |
| 3 Dst IP object | |
| 4 Dst Port int64 | |
| 5 Protocol float64 | |
| 6 Timestamp object | |
| 7 Flow Duration float64 | |
| 8 Total Fwd Packet float64 | |
| 9 Total Bwd packets float64 | |
| 10 Total Length of Fwd Packet float64 | |
| 11 Total Length of Bwd Packet float64 | |
| 12 Fwd Packet Length Max float64 | |
| 13 Fwd Packet Length Min float64 | |
| 14 Fwd Packet Length Mean float64 | |
| 15 Fwd Packet Length Std float64 | |
| 16 Bwd Packet Length Max float64 | |
| 17 Bwd Packet Length Min float64 | |
| 18 Bwd Packet Length Mean float64 | |
| 19 Bwd Packet Length Std float64 | |
| 20 Flow Bytes/s float64 | |
| 21 Flow Packets/s float64 | |
| 22 Flow IAT Mean float64 | |
| 23 Flow IAT Std float64 | |
| 24 Flow IAT Max float64 | |
| 25 Flow IAT Min float64 | |
| 26 Fwd IAT Total float64 | |
| 27 Fwd IAT Mean float64 | |
| 28 Fwd IAT Std float64 | |
| 29 Fwd IAT Max float64 | |
| 30 Fwd IAT Min float64 | |
| 31 Bwd IAT Total float64 | |
| 32 Bwd IAT Mean float64 | |
| 33 Bwd IAT Std float64 | |
| 34 Bwd IAT Max float64 | |
| 35 Bwd IAT Min float64 | |
| 36 Fwd PSH Flags float64 | |
| 37 Bwd PSH Flags float64 | |
| 38 Fwd URG Flags float64 | |
| 39 Bwd URG Flags float64 | |
| 40 Fwd Header Length float64 | |
| 41 Bwd Header Length float64 | |
| 42 Fwd Packets/s float64 | |
| 43 Bwd Packets/s float64 | |
| 44 Packet Length Min float64 | |
| 45 Packet Length Max float64 | |
| 46 Packet Length Mean float64 | |
| 47 Packet Length Std float64 | |
| 48 Packet Length Variance float64 | |
| 49 FIN Flag Count float64 | |
| 50 SYN Flag Count float64 | |
| 51 RST Flag Count float64 | |
| 52 PSH Flag Count float64 | |
| 53 ACK Flag Count float64 | |
| 54 URG Flag Count float64 | |
| 55 CWR Flag Count float64 | |
| 56 ECE Flag Count float64 | |
| 57 Down/Up Ratio float64 | |
| 58 Average Packet Size float64 | |
| 59 Fwd Segment Size Avg float64 | |
| 60 Bwd Segment Size Avg float64 | |
| 61 Fwd Bytes/Bulk Avg float64 | |
| 62 Fwd Packet/Bulk Avg float64 | |
| 63 Fwd Bulk Rate Avg float64 | |
| 64 Bwd Bytes/Bulk Avg float64 | |
| 65 Bwd Packet/Bulk Avg float64 | |
| 66 Bwd Bulk Rate Avg float64 | |
| 67 Subflow Fwd Packets float64 | |
| 68 Subflow Fwd Bytes float64 | |
| 69 Subflow Bwd Packets float64 | |
| 70 Subflow Bwd Bytes float64 | |
| 71 FWD Init Win Bytes float64 | |
| 72 Bwd Init Win Bytes float64 | |
| 73 Fwd Act Data Pkts float64 | |
| 74 Fwd Seg Size Min float64 | |
| 75 Active Mean float64 | |
| 76 Active Std float64 | |
| 77 Active Max float64 | |
| 78 Active Min float64 | |
| 79 Idle Mean float64 | |
| 80 Idle Std float64 | |
| 81 Idle Max float64 | |
| 82 Idle Min float64 | |
| 83 Label object | |
| 84 Traffic Type object | |
| 85 Traffic Subtype object | |
| dtypes: float64(78), int64(1), object(7) | |
| memory usage: 5.5+ GB | |
|  | |

Όπως φαίνεται, συνολικά υπάρχουν 86 στήλες τύπων int64, float64 και object. Υπάρχουν επίσης, 8656767 δείγματα και η χρήση της μνήμης είναι στα 5.5+ GB.

### Συγκεντρωτικά Στατιστικά Μεγέθη

Τα συγκεντρωτικά στατιστικά μεγέθη υπολογίζονται μέσω της συνάρτησης describe() της pandas και αποθηκεύονται στον folder “data” ως statistics.csv. Απόσπασμα του περιεχομένου του αρχείου φαίνεται παρακάτω.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Flow ID | Src IP | Src Port | Dst IP | Dst Port | Protocol |
| count | 8656767 | 8656767 | 8656767 | 8656767 | 8656767 | 8656767 |
| unique | 951935 | 13 |  | 15 |  |  |
| top | 192.168.1.59-192.168.1.177-2323-65055-6 | 192.168.1.70 | | 192.168.1.90 | |  |
| freq | 904 | 8540951 |  | 8540909 |  |  |
| mean |  |  | 25633.13 |  | 4050.301 | 6.426944 |
| std |  |  | 20059.25 |  | 12685.83 | 2.12567 |
| min |  |  | 0 |  | 0 | 0 |
| 25% |  |  | 6366 |  | 0 | 6 |
| 50% |  |  | 22240 |  | 0 | 6 |
| 75% |  |  | 42342 |  | 0 | 6 |
| max |  |  | 65535 |  | 65535 | 17 |
|  |  |  |  |  |  |  |

Στο αρχείο φαίνεται το πλήθος των στοιχείων των στηλών, το πλήθος των μοναδικών τιμών, η πιο συχνά εμφανιζόμενη τιμή και η συχνότητά της, η ενδιάμεση τιμή, η τυπική απόκλιση, η ελάχιστη και η μέγιστη τιμή, όπως και τα τεταρτημόρια.

### Γραφικές Αναπαραστάσεις

Οι γραφικές παραστάσεις που δημιουργούνται εξαρτώνται από το είδος των δεδομένων της κάθε στήλης. Πιο συγκεκριμένα, για τις κατηγορικές στήλες, δημιουργούνται Bar Plots που απεικονίζουν τις πιο συχνές τιμές (εάν αυτές είναι πάνω από 100 σε πλήθος, τότε απεικονίζονται μόνο οι 20 πιο συχνές για εξοικονόμηση μνήμης). Για τις αριθμητικές στήλες, δημιουργούνται Histograms που απεικονίζουν την κατανομή των τιμών τους, αλλά και Boxplots που δείχνουν την κατανομή και διασπορά της κάθε αριθμητικής στήλης. Τα διαγράμματα αυτά δεν μπορούν όλα να περιληφθούν στην αναφορά λόγο του πλήθους του (υπάρχουν 10 Bar plots, 76 Histograms και 76 Boxplots), οπότε θα παρουσιαστούν συνοπτικά τα συμπεράσματα που προκύπτουν από αυτά.

* Στα bar plots φαίνεται ότι υπάρχει πολύ άνιση κατανομή των κατηγορικών δεδομένων. Για παράδειγμα, η στήλη Label περιέχει περίπου δέκα εκατομμύρια εγγραφές του τύπου Malicious και μόνο 100 του τύπου Bening.

A blue rectangular bar with white text

AI-generated content may be incorrect.

Παρόμοιες κατανομές παρατηρούνται και στις στήλες Dst\_IP, Dst\_Port και σε άλλες στήλες ακόμα.

* Τα boxplots στην συνέχεια είναι λίγο πιο δύσκολο στην κατανόηση τους. Η πλειοψηφία των boxplots έχει την εξής μορφή:

A graph with a line graph

AI-generated content may be incorrect.

Λόγο του μεγάλου πλήθος των εγγραφών, δεν μπορεί εύκολα να αναπαρασταθεί ολόκληρη η πληροφορία. Αυτό που κατανοούμε είναι πως οι περισσότερες τιμές τις κάθε αριθμητικής στήλης είναι κοντά μεταξύ τους, αλλά υπάρχουν και συγκεκριμένες τιμές που απέχουν αρκετά από την μέση τιμή. Εξαίρεση αποτελούν οι στήλες Bwd\_PSH\_Flags και Bwd\_URG\_Flags που παρουσιάζουν μηδενική απόκλιση από την επικρατούσα τιμή.

* Τέλος, τα histograms παρουσιάζουν την ίδια κατανομή με στην περίπτωση των bar plots.

A graph with a bar graph

AI-generated content may be incorrect.

Που δείχνουν ότι οι πλειοψηφία των τιμών των αριθμητικών στηλών πέφτουν σε ένα συγκεκριμένο εύρος, και τα outliers είναι λίγα.

Από τις παραπάνω γραφικές παραστάσεις και από τα συγκεντρωτικά στατιστικά μεγέθη, γίνεται αντιληπτό πως το csv δεν πρόκειται για ένα ομοιόμορφο αρχείο όπου κάθε στήλη έχει παρόμοιο πλήθος εμφανίσεων της κάθε τιμής, αλλά πρόκειται για ένα αρχείο όπου σε αρκετές στήλες οι τιμές που επικρατούν είναι λίγες.

### Συσχετίσεις

Οι συσχετίσεις υπολογίζονται μέσω της συνάρτησης corr() της pandas σε δύο διαφορετικές περιστάσεις. Αρχικά πάνω σε ολόκληρο το dataset “data.csv” και έπειτα πάνω στα στατιστικά “statistics.csv”. Από αυτά προκύπτει το μητρώο συσχέτισης και στη συνέχεια κάθε ζεύγος στήλης-γραμμής του μητρώου διατάσσεται ως προς φθίνουσα συσχέτιση. Τα δύο αρχεία που προκύπτουν, αποθηκεύονται στον folder “data” και λέγονται “column\_correlation\_matrix.xlsx” και “statistics\_correlation\_matrix.xlsx”. Απόσπασμα του μητρώου του πρώτου αρχείου είναι το παρακάτω.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Flow Duration** | **Total Fwd Packet** | **Total Bwd packets** | **Total Length of Fwd Packet** | **Total Length of Bwd Packet** | **Fwd Packet Length Max** | **Fwd Packet Length Min** | **Fwd Packet Length Mean** | **Fwd Packet Length Std** |
| **Flow Duration** | 1 | 0.014302 | 0.00662 | 0.022835 | 0.006789 | 0.28237 | 0.199538 | 0.235432 | 0.317529 |
| **Total Fwd Packet** | 0.014302 | 1 | 0.334657 | 0.061667 | 0.028327 | -0.0013 | -0.00174 | -0.00166 | 0.000684 |
| **Total Bwd packets** | 0.00662 | 0.334657 | 1 | 0.002402 | 0.217519 | -0.00102 | -0.00248 | -0.00244 | 0.00104 |
| **Total Length of Fwd Packet** | 0.022835 | 0.061667 | 0.002402 | 1 | 0.001303 | 0.02968 | 0.028722 | 0.029312 | 0.004577 |
| **Total Length of Bwd Packet** | 0.006789 | 0.028327 | 0.217519 | 0.001303 | 1 | -0.00172 | -0.00407 | -0.00401 | 0.002377 |
| **Fwd Packet Length Max** | 0.28237 | -0.0013 | -0.00102 | 0.02968 | -0.00172 | 1 | 0.962844 | 0.984035 | 0.198755 |
| **Fwd Packet Length Min** | 0.199538 | -0.00174 | -0.00248 | 0.028722 | -0.00407 | 0.962844 | 1 | 0.99451 | -0.0644 |
| **Fwd Packet Length Mean** | 0.235432 | -0.00166 | -0.00244 | 0.029312 | -0.00401 | 0.984035 | 0.99451 | 1 | 0.035167 |
| **Fwd Packet Length Std** | 0.317529 | 0.000684 | 0.00104 | 0.004577 | 0.002377 | 0.198755 | -0.0644 | 0.035167 | 1 |

Απόσπασμα των ταξινομημένων ως προς τη συσχέτιση ζευγών είναι το παρακάτω.

|  |  |  |  |
| --- | --- | --- | --- |
| **Variable 1** | **Variable 2** | **Correlation** | **Absolute Correlation** |
| Bwd Packet Length Mean | Bwd Segment Size Avg | 1 | 1 |
| Flow Duration | Flow Duration | 1 | 1 |
| Flow Packets/s | Fwd Packets/s | 0.999946 | 0.999946 |
| Fwd Packet Length Max | Packet Length Max | 0.997144 | 0.997144 |
| Fwd Packet Length Min | Fwd Packet Length Mean | 0.99451 | 0.99451 |
| Flow IAT Max | Idle Max | 0.994274 | 0.994274 |

Αντίστοιχα το αρχείο “statistics\_correlation\_matrix.xlsx” στην πρώτη σελίδα περιέχει το μητρώο:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | **mean** | **std** | **min** | **25%** | **50%** | **75%** | **max** |
| **mean** | 1 | 0.970329 | -0.10697 | 0.548586 | 0.828101 | 0.991255 | -0.01246 |
| **std** | 0.970329 | 1 | -0.16904 | 0.53175 | 0.775734 | 0.953271 | 0.196126 |
| **min** | -0.10697 | -0.16904 | 1 | -0.31562 | 0.024171 | -0.05892 | 0.004024 |
| **25%** | 0.548586 | 0.53175 | -0.31562 | 1 | 0.579526 | 0.508659 | -0.00706 |
| **50%** | 0.828101 | 0.775734 | 0.024171 | 0.579526 | 1 | 0.803441 | -0.0089 |
| **75%** | 0.991255 | 0.953271 | -0.05892 | 0.508659 | 0.803441 | 1 | -0.0118 |
| **max** | -0.01246 | 0.196126 | 0.004024 | -0.00706 | -0.0089 | -0.0118 | 1 |

Στη δεύτερη σελίδα είναι τα ζεύγη ταξινομημένα.

|  |  |  |  |
| --- | --- | --- | --- |
| **Variable 1** | **Variable 2** | **Correlation** | **Absolute Correlation** |
| mean | 75% | 0.991255 | 0.991255 |
| mean | std | 0.970329 | 0.970329 |
| std | 75% | 0.953271 | 0.953271 |
| mean | 50% | 0.828101 | 0.828101 |
| 50% | 75% | 0.803441 | 0.803441 |
| std | 50% | 0.775734 | 0.775734 |
| 25% | 50% | 0.579526 | 0.579526 |
| mean | 25% | 0.548586 | 0.548586 |
| std | 25% | 0.53175 | 0.53175 |
| 25% | 75% | 0.508659 | 0.508659 |
| min | 25% | -0.31562 | 0.315624 |
| std | max | 0.196126 | 0.196126 |
| std | min | -0.16904 | 0.16904 |
| mean | min | -0.10697 | 0.106973 |
| min | 75% | -0.05892 | 0.058924 |
| min | 50% | 0.024171 | 0.024171 |
| mean | max | -0.01246 | 0.012463 |
| 75% | max | -0.0118 | 0.011796 |
| 50% | max | -0.0089 | 0.008901 |
| 25% | max | -0.00706 | 0.007058 |
| min | max | 0.004024 | 0.004024 |

* Από τις ταξινομημένες συσχετίσεις του αρχικού dataset φαίνεται να έχουν μεγάλη συσχέτιση, πάνω από 99% οι στήλες:

|  |  |
| --- | --- |
| Bwd Packet Length Mean | Bwd Segment Size Avg |
| Flow Duration | Flow Duration |
| Flow Packets/s | Fwd Packets/s |
| Fwd Packet Length Max | Packet Length Max |
| Fwd Packet Length Min | Fwd Packet Length Mean |
| Flow IAT Max | Idle Max |
| Packet Length Mean | Average Packet Size |
| Bwd Bytes/Bulk Avg | Bwd Packet/Bulk Avg |

* Στο αρχείο των στατιστικών, από την άλλη φαίνεται να έχουν μεγάλη συσχέτιση τα στατιστικά:

mean 75%

mean std

std 75%

# Ερώτημα 2: Μείωση Συνόλου Δεδομένων

## Μείωση στηλών

Το πρώτο βήμα για την μείωση της διάστασης του αρχείου, είναι να εξετάσουμε ποιες από τις αριθμητικές στήλες εμφανίζουν υψηλή συσχέτιση. Ανατρέχοντας στην δεύτερη σελίδα του “column\_correlation\_matrix.xlsx ”, μπορούμε να δούμε τις συσχετίσεις των στηλών με φθίνουσα κατάταξη. Όπως φαίνεται στο κάτω μέρος της σελίδας 10, αυτές είναι οι πρώτες στήλες του αρχείου:

|  |  |  |  |
| --- | --- | --- | --- |
| **Variable 1** | **Variable 2** | **Correlation** | **Absolute Correlation** |
| Bwd Packet Length Mean | Bwd Segment Size Avg | 1 | 1 |
| Flow Duration | Flow Duration | 1 | 1 |
| Flow Packets/s | Fwd Packets/s | 0.999946 | 0.999946 |
| Fwd Packet Length Max | Packet Length Max | 0.997144 | 0.997144 |
| Fwd Packet Length Min | Fwd Packet Length Mean | 0.99451 | 0.99451 |
| Flow IAT Max | Idle Max | 0.994274 | 0.994274 |

Από τις στήλες που εμφανίζουν υψηλή συσχέτιση (μεγαλύτερη του 90%), διατηρούμε μόνο την μία ενώ κάνουμε drop την άλλη. Για παράδειγμα, από τις στήλες Flow Packets/s και Fwd Packets/s, κάνουμε drop την Fwd Packets/s και διατηρούμε μόνο την Flow Packets/s. Την διαδικασία αυτή την επαναλαμβάνουμε για όλες της στήλες με correlation μεγαλύτερη του 90%. Επίσης κάνουμε drop μερικές στήλες με μικρή σημασία, όπως είναι για παράδειγμα η Timestamp. Τελικά, οι στήλες που γίνονται drop είναι οι παρακάτω:

# Columns to drop

columns\_to\_drop = [

    'Timestamp', 'Bwd Segment Size Avg', 'Fwd Packets/s', 'Packet Length Max',

    'Fwd Packet Length Mean', 'Flow IAT Max', 'Packet Length Mean',

    'Bwd Packet/Bulk Avg', 'Active Min', 'ACK Flag Count', 'Active Max',

    'Idle Min', 'Bwd IAT Total', 'Fwd Act Data Pkts', 'Fwd IAT Max',

    'Average Packet Size', 'Fwd Packet Length Min', 'Subflow Bwd Packets',

    'Bwd Packet Length Std', 'Bwd IAT Max', 'Fwd IAT Mean', 'Idle Mean',

    'Packet Length Variance'

]

## Δειγματοληψία

Το επόμενο βήμα στην μείωση της διάστασης του αρχείου είναι η μείωση του πλήθους των εγγραφών. Για να το επιτύχουμε αυτό, εφαρμόζουμε 3 τεχνικές:

1. Stratified Δειγματοληψία.
2. Clustering με kmeans.
3. Clustering με birch.

Η δειγματοληψία γίνεται εύκολα με την εξής εντολή:

df\_stratified = df.groupby(['Label', 'Traffic Type'], group\_keys=False).sample(frac=sample\_frac, random\_state=42)

Το group by γίνεται με βάση τις στήλες Label και Traffic Type γιατί αυτές μας ενδιαφέρουν για το τρίτο ερώτημα που αφορά την εκπαίδευση. Το csv που προκύπτει από την δειγματοληψία είναι το *¨data\_stratified.csv ¨*.

Στην συνέχεια εφαρμόζουμε scaling των στηλών για το clustering που ακολουθεί.

Ο Kmeans δημιουργεί clusters από τις εγγραφές του αρχικού αρχείου και έπειτα παίρνει 2 τοις εκατό των εγγραφών από το κάθε cluster ώστε να δημιουργηθεί ένα μικρότερο αρχείο csv, με τίτλο *¨data\_kmeans\_custom.csv¨*. Ωστόσο, εδώ αντιμετωπίσαμε ένα σημαντικό πρόβλημα. Η στήλη Label έχει περίπου 100 εγγραφές από την κλάση Bening και περίπου 10 εκατομμύρια από την κλάση Malicious. Άρα το δύο τοις εκατό από το cluster με Label Bening, επιστρέφει δύο μόνο εγγραφές το οποίο δεν είναι αρκετό για την εκπαίδευση του μοντέλου στο ερώτημα 3. Κατά τον πειραματισμό με το clustering, παρατηρήθηκε ότι το cluster στο οποίο καταλήγουν οι εγγραφές με Label Bening ήταν το cluster 1. Για αυτό τον λόγο, κατά την δειγματοληψία των clusters, παίρνουμε περισσότερες εγγραφές από το cluster 1 που είναι το μικρότερο και θέλουμε περισσότερες εγγραφές από αυτό και λιγότερες από τα υπόλοιπα.

for cluster\_id, group in df.groupby('Cluster'):

    if cluster\_id == 1:

        samples.append(group.sample(frac=0.5, random\_state=42))  # take half of cluster 1

    else:

        samples.append(group.sample(frac=sample\_frac, random\_state=42))  # 2% from others

Συγκεκριμένα, παίρνουμε το μισό από το cluster με id = 1 και 2 % από τα υπόλοιπα.

Για το δεύτερο clustering χρησιμοποιήθηκε ο birch, αλλά όχι στο αρχικό dataset αλλά σε αυτό που προέκυψε κατά την δειγματοληψία. Αυτό έγινε όχι από ανάγκης ram (το υπολογιστικό σύστημα που έτρεχε ο κώδικας είχε 32 gb ram), αλλά επειδή έπαιρνε πολύ ώρα να τερματίσει ο birch, όπως και άλλοι clustering αλγόριθμοι που δοκιμάστηκαν, εξαιτίας του μεγάλου αρχικού dataset. Στον birch, πάλι παίρνουμε το 50 % από το cluster με id 2 που είναι το cluster όπου το Label είναι benign και είναι μικρό σε μέγεθος, ενώ 2 % από τα υπόλοιπα clusters, ώστε να επιτύχουμε καλύτερο training στην συνέχεια. Έτσι δημιουργείται το *¨data\_birch\_custom.csv¨*.

# Ερώτημα 3

## Περιγραφή Υλοποίησης

Για την υλοποίηση του ερωτήματος χρησιμοποιήθηκαν δύο μοντέλα της sklearn, το SVC και το MLP που υλοποιούν το κλασικό Support Vector Machine και Multilayer Perceptron αντίστοιχα. Για την ταξινόμηση χρησιμοποιούνται δύο στήλες του dataset. Η “Label” που έχει δύο μόνο τιμές (Benign και Malicious) και κατά συνέπεια καθιστά το πρόβλημα binary classification και η “Traffic Type” που έχει περισσότερες των δύο τιμές και καθιστά το πρόβλημα multiclass classification. Κατά το τρέξιμο το πρόγραμμα ζητά από το χρήστη, αρχικά να διαλέξει ένα από τα τρία αρχεία που προέκυψαν από το ερώτημα 2 (data\_stratified.csv, data\_kmeans\_custom.csv, data\_birch\_custom.csv). Στη συνέχεια ζητά τη στήλη που θα αποτελέσει την ετικέτα κλάσεων ( binary vs multiclass classification) και τέλος, ζητά τον ταξινομητή, δηλαδή MLP ή SVM.

Αφού ο χρήστης εισαγάγει τιμές, η στήλη που στο εκάστοτε τρέξιμο αποτελεί την ετικέτα κλάσης περνάει μέσα από label encoder για να μετατραπούν οι τιμές της σε αριθμητικές, καθώς τα μοντέλα που πρόκειται να χρησιμοποιηθούν διαχειρίζονται μόνο αριθμητικά δεδομένα. Στη συνέχεια, τα features χωρίζονται σε αριθμητικά και κατηγορικά. Τα κατηγορικά περνούν μέσα από feature hasher (MurmurHash-based για ταχύτητα και καλή διαχείριση μνήμης) που τα κωδικοποιεί ως αριθμητικές τιμές, ενώ τα αριθμητικά περνούν μέσα από standard scaler για να κανονικοποιηθούν και να είναι αποδοτικότερη η ταξινόμηση.

Αφού τελειώσει το στάδιο της προεπεξεργασίας, ακολουθεί η εκπαίδευση των μοντέλων. Για μεγαλύτερη ταχύτητα, λόγω πολλών δεδομένων αλλά και πολλών διαφορετικών τρεξιμάτων λόγω tests διαφορετικών μοντέλων/αρχείων/κλάσεων, δε χρησιμοποιούνται χρονοβόρες μέθοδοι όπως leave one out ή k-fold cross validation, αλλά ένα απλό train-test split με αναλογία 80-20. Οι παράμετροι που χρησιμοποιούνται κατά την αρχικοποίηση των μοντέλων εστιάζουν κυρίως στην ταχύτητα, προφανώς σε βάρος της απόδοσής τους.

Τέλος, για την αξιολόγηση χρησιμοποιούνται οι συναρτήσεις predict() και classification\_report(). Η τελευταία εμφανίζει αναλυτικά τα αποτελέσματα της αξιολόγησης για το κάθε τρέξιμο. Τα αποτελέσματα παρατίθενται παρακάτω.

## Αξιολόγηση

### Sampled Data, Binary Classification, MLP

Classification Report:

precision recall f1-score support

0 1.00 0.17 0.29 6

1 1.00 1.00 1.00 34621

accuracy 1.00 34627

macro avg 1.00 0.58 0.64 34627

weighted avg 1.00 1.00 1.00 34627

### Sampled Data, Binary Classification, SVM

Classification Report:

precision recall f1-score support

0 0.14 0.17 0.15 6

1 1.00 1.00 1.00 34621

accuracy 1.00 34627

macro avg 0.57 0.58 0.58 34627

weighted avg 1.00 1.00 1.00 34627

### Sampled Data, Multiclass Classification, MLP

Classification Report:

precision recall f1-score support

0 1.00 1.00 1.00 1

2 0.98 0.98 0.98 155

3 1.00 1.00 1.00 29894

4 1.00 1.00 1.00 4192

5 0.99 0.77 0.86 380

7 0.00 0.00 0.00 5

accuracy 1.00 34627

macro avg 0.83 0.79 0.81 34627

weighted avg 1.00 1.00 1.00 34627

### Sampled Data, Multiclass Classification, SVM

Classification Report:

precision recall f1-score support

0 0.00 0.00 0.00 1

2 0.79 0.43 0.55 155

3 0.98 0.94 0.96 29894

4 0.96 0.47 0.63 4192

5 0.03 0.32 0.06 380

6 0.00 0.00 0.00 0

7 0.50 0.40 0.44 5

accuracy 0.87 34627

macro avg 0.46 0.36 0.38 34627

weighted avg 0.96 0.87 0.90 34627

### K-means Data, Binary Classification, MLP

Classification Report:

precision recall f1-score support

0 1.00 0.89 0.94 19

1 1.00 1.00 1.00 34621

accuracy 1.00 34640

macro avg 1.00 0.95 0.97 34640

weighted avg 1.00 1.00 1.00 34640

### K-means Data, Binary Classification, SVM

Classification Report:

precision recall f1-score support

0 0.32 0.89 0.47 19

1 1.00 1.00 1.00 34621

accuracy 1.00 34640

macro avg 0.66 0.95 0.74 34640

weighted avg 1.00 1.00 1.00 34640

### K-means Data, Multiclass Classification, MLP

Classification Report:

precision recall f1-score support

0 1.00 0.90 0.95 10

1 0.96 0.98 0.97 131

2 1.00 1.00 1.00 30008

3 1.00 1.00 1.00 4117

4 0.96 0.94 0.95 365

5 1.00 0.50 0.67 2

6 1.00 1.00 1.00 7

accuracy 1.00 34640

macro avg 0.99 0.90 0.93 34640

weighted avg 1.00 1.00 1.00 34640

### K-means Data, Multiclass Classification, SVM

Classification Report:

precision recall f1-score support

0 1.00 1.00 1.00 10

1 0.42 0.40 0.41 131

2 0.99 0.93 0.96 30008

3 0.98 0.58 0.73 4117

4 0.02 0.16 0.03 365

5 1.00 0.50 0.67 2

6 0.88 1.00 0.93 7

accuracy 0.88 34640

macro avg 0.75 0.65 0.67 34640

weighted avg 0.97 0.88 0.92 34640

### Birch Data, Binary Classification, MLP

Classification Report:

precision recall f1-score support

0 0.33 0.50 0.40 2

1 1.00 1.00 1.00 17312

accuracy 1.00 17314

macro avg 0.67 0.75 0.70 17314

weighted avg 1.00 1.00 1.00 17314

### Birch Data, Binary Classification, SVM

Classification Report:

precision recall f1-score support

0 0.25 0.50 0.33 2

1 1.00 1.00 1.00 17312

accuracy 1.00 17314

macro avg 0.62 0.75 0.67 17314

weighted avg 1.00 1.00 1.00 17314

### Birch Data, Multiclass Classification, MLP

Classification Report:

precision recall f1-score support

1 0.98 0.98 0.98 86

2 1.00 1.00 1.00 14961

3 1.00 1.00 1.00 2086

4 0.89 0.75 0.81 179

5 1.00 1.00 1.00 1

6 0.00 0.00 0.00 1

accuracy 1.00 17314

macro avg 0.81 0.79 0.80 17314

weighted avg 1.00 1.00 1.00 17314

### Birch Data, Multiclass Classification, SVM

Classification Report:

precision recall f1-score support

0 0.00 0.00 0.00 0

1 0.69 0.55 0.61 86

2 1.00 0.92 0.96 14961

3 0.99 0.83 0.90 2086

4 0.08 0.80 0.15 179

5 1.00 1.00 1.00 1

6 0.00 0.00 0.00 1

accuracy 0.90 17314

macro avg 0.54 0.59 0.52 17314

weighted avg 0.99 0.90 0.94 17314

## Σχολιασμός Αποτελεσμάτων

Από τα παραπάνω τρεξίματα προκύπτουν οι εξής παρατηρήσεις:

* Τα SVMs τρέχουν πιο αργά από τα MLPs (με ένα ή δύο layers).
* Τα MLPs (με ένα ή δύο layers) κάνουν καλύτερο classification στο dataset από τα SVMs, χωρίς όμως μεγάλη διαφορά.
* Το binary classification, με εξαίρεση την περίπτωση των δεδομένων του K-means, ταξινομούσε καλά την μεγάλη κλάση, αλλά όχι τη μικρή. Αυτό οφείλεται στην κατανομή των δεδομένων. Υπήρχαν πολύ περισσότερα δείγματα στη μεγάλη κλάση και τα μοντέλα μάθαιναν καλύτερα να τα αναγνωρίζουν.
* Το multiclass classification ταξινομούσε καλύτερα τις μεγάλες και μεσαίες κλάσεις.
* Καλύτερα αποτελέσματα έχει το αρχείο K-means data.
* Καλύτερος συνδυασμός φαίνεται να είναι ο K-means data με MLP.