|  |  |
| --- | --- |
| pyrforos.png | **ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ**  ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ |

**ΤΕΧΝΟΛΟΓΙΕΣ ΚΙΝΗΤΟΥ ΥΠΟΛΟΓΙΣΜΟΥ ΜΕ ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ**

**ΧΕΙΜΕΡΙΝΟ ΕΞΑΜΗΝΟ** **-**

**Η ΕΡΓΑΣΤΗΡΙΑΚΗ ΑΣΚΗΣΗ**

**ΑΞΙΟΛΟΓΗΣΗ ΑΛΓΟΡΙΘΜΩΝ ΜΗΧΑΝΙΚΗΣ ΜΑΘΗΣΗΣ ΓΙΑ ΤΗΝ ΠΡΟΒΛΕΨΗ ΤΟΥ ΡΥΘΜΟΥ ΔΙΕΛΕΥΣΗΣ (THROUGHPUT) ΣΕ ΔΙΚΤΥΑ 𝟓G**

**Παναγιώτης-Αλέξιος Σπανάκης 03400274**

**Ιωακείμ Ελ-Χαττάμπ-Μπριστογιάννης 03400249**

**04/01/2025**

## Εισαγωγή

Σκοπός της παρακάτω εργασίας ήταν, η αξιολόγηση αλγορίθμων μηχανικής μάθησης για την πρόβλεψη του ρυθμού διέλευσης (**throughput**) σε δίκτυα 5G.

Τα δεδομένα δημιουργήθηκαν, με χρήση προσομοιωτών (βασισμένων στα πρότυπα ETSI TR 38.901) , στο Εργαστήριο Ευφυών Επικοινωνιών και Δικτύων Ευρείας Ζώνης (ICBNet) και στο Εργαστήριο Μικροκυμάτων και Οπτικών Ινών (MFOL). Από τους παραπάνω προσομοιωτές δημιουργούνται σύνολα δεδομένων, με σκοπό την πρόβλεψη του ρυθμού διέλευσης (throughput) που επιτυγχάνεται για τον εκάστοτε χρήστη που εισέρχεται στην τοπολογία, τα οποία αποτελούνται από 10075 δείγματα έχουν τα ακόλουθα χαρακτηριστικά (features) για κάθε χρήστη:

• 𝑥, 𝑦 συντεταγμένες του χρήστη στην τοπολογία, όπου 𝑥, 𝑦 ∈ [−1800, 1800]

• σταθμό βάσης (BS) εξυπηρέτησης

• τομέα (sector) εξυπηρέτησης

• είδος διαμόρφωσης (QPSK, 16-QAM, 64-QAM)

• ισχύ σήματος

• απόσταση χρήστη - BS εξυπηρέτησης

• γωνιακή απόσταση χρήστη - BS εξυπηρέτησης

• ρυθμό διέλευσης (throughput)

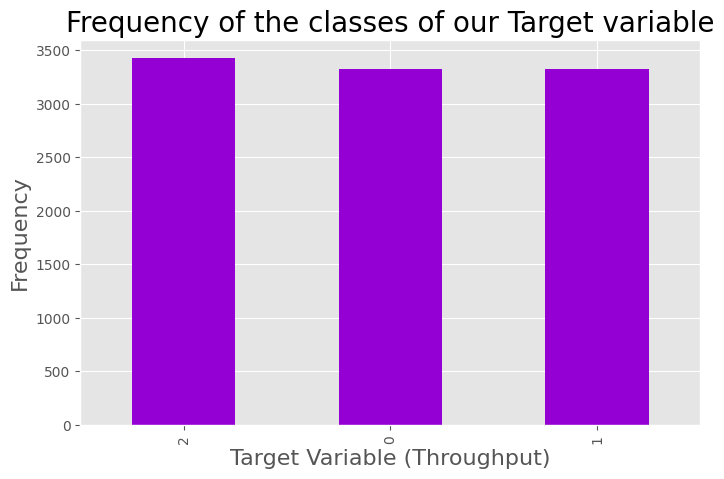
Τα 7 πρώτα features λειτουργούν ως μεταβλητές πρόβλεψης (predictor variables), ενώ ο ρυθμός διέλευσης λειτουργεί ως μεταβλητή απόκρισης (response variable).

**Προεπεξεργασία Δεδομένων**

Η διαδικασία προεπεξεργασίας δεδομένων ακολούθησε μια σειρά από στοχευμένα βήματα για την εξασφάλιση της ποιότητας και της καταλληλότητας του συνόλου δεδομένων, προτού προχωρήσουμε στην εφαρμογή του Chi-Squared Feature Selection. Ακολουθεί αναλυτική περιγραφή κάθε σταδίου:

1. **Διαχωρισμός της Μεταβλητής Στόχου ("throughput")**

Για την περίπτωση του classification η μεταβλητή στόχος, "throughput", διαχωρίστηκε σε τρεις κατηγορίες μέσω τεχνικής διαμέρισης ισομεγέθων ομάδων (equal frequency binning). Η επιλογή αυτή έγινε με σκοπό να μετατρέψουμε ένα συνεχές πρόβλημα σε κατηγορικό. Οι τρεις κατηγορίες περιείχαν ίσο αριθμό δειγμάτων, ώστε να αποφευχθούν ανισορροπίες που θα μπορούσαν να επηρεάσουν την ανάλυση.



1. **Εξερεύνηση της Κατανομής των Αριθμητικών Μεταβλητών**

Πραγματοποιήθηκε διερευνητική ανάλυση δεδομένων (EDA) για την καταγραφή της κατανομής των αριθμητικών χαρακτηριστικών. Δημιουργήθηκαν γραφήματα συχνότητας για τα κατηγορικά και ψευτοκατηγορικά χαρακτηριστικά (αριθμητικά χαρακτηριστικά με μικρό πλήθος τιμών).

Εικόνα που περιέχει κείμενο, στιγμιότυπο οθόνης, διάγραμμα, ορθογώνιο παραλληλόγραμμο

Περιγραφή που δημιουργήθηκε αυτόματα

Για τα υπόλοιπα χαρακτηριστικά όπως δημιουργήσαμε 3 διαγράμματα distribution, q-q και box plot, προκειμένου να εντοπιστούν πιθανά outliers, καθώς και μοτίβα που σχετίζονται με τη μεταβλητή στόχο. Τα γραφήματα αυτά έδειξαν ότι οι περισσότερες αριθμητικές μεταβλητές παρουσίαζαν ασύμμετρη κατανομή, χωρίς ακραίες τιμές που να απαιτούν περαιτέρω διόρθωση.

Εικόνα που περιέχει γράφημα, διάγραμμα, στιγμιότυπο οθόνης, γραμμή

Περιγραφή που δημιουργήθηκε αυτόματα

1. **Μετατροπή Κατηγορικών Δεδομένων**

Για τη μετατροπή των κατηγορικών και ψευδοκατηγορικών δεδομένων σε μορφή κατάλληλη για μαθηματική επεξεργασία, εφαρμόστηκε ο Label Encoder. Με τον τρόπο αυτό, οι κατηγορικές μεταβλητές αντιστοιχίστηκαν σε ακέραιες τιμές, διατηρώντας τη διακριτότητα των κατηγοριών, ενώ αποφεύχθηκε η δημιουργία υπερβολικά πολλών χαρακτηριστικών, όπως συμβαίνει με την τεχνική One-Hot Encoding.

1. **Δημιουργία Νέων Χαρακτηριστικών**

Για την εμπλουτισμό του συνόλου δεδομένων, δημιουργήθηκαν τα ακόλουθα σύνθετα χαρακτηριστικά:

* Distance\_From\_Origin: Η απόσταση κάθε χρήστη από την αρχή των συντεταγμένων.
* Power\_Distance\_Ratio: Ο λόγος ισχύος προς απόσταση, κρίσιμος για την ανάλυση της απόδοσης.
* BS\_UEAngle\_sin, BS\_UEAngle\_cos: Τριγωνομετρικοί μετασχηματισμοί της γωνίας μεταξύ σταθμού βάσης και χρήστη, οι οποίοι επιτρέπουν την κυκλική αναπαράσταση των δεδομένων.
* BS\_UEAngle\_degrees: Η γωνία σε μοίρες για ερμηνευτικούς σκοπούς.
* Power\_Distance\_Interaction, Modulation\_Power\_Interaction: Διαδραστικά χαρακτηριστικά που συνδυάζουν τις ισχύουσες παραμέτρους για τον εντοπισμό μη γραμμικών σχέσεων.

Η δημιουργία αυτών των χαρακτηριστικών βασίστηκε σε θεμελιώδεις αρχές του 5G δικτύου, εξασφαλίζοντας ότι ενσωματώνεται φυσική και τεχνική γνώση στο μοντέλο.

1. **Αφαίρεση Χαρακτηριστικών με Υψηλή Συσχέτιση**

Η συσχέτιση μεταξύ χαρακτηριστικών εξετάστηκε μέσω του συντελεστή Pearson. Χαρακτηριστικά με συσχέτιση άνω του 0.6 θεωρήθηκαν πλεονασματικά, καθώς η παρουσία τους θα μπορούσε να επηρεάσει αρνητικά την απόδοση του μοντέλου. Σε κάθε ζεύγος χαρακτηριστικών με υψηλή συσχέτιση, διατηρήθηκε εκείνο που παρουσίαζε υψηλότερη σημασία για τη μεταβλητή στόχο.

Εικόνα που περιέχει κείμενο, στιγμιότυπο οθόνης, τετράγωνο, ορθογώνιο παραλληλόγραμμο

Περιγραφή που δημιουργήθηκε αυτόματα

Η συσχέτιση των χαρακτηριστικών πριν την αφαίρεση

Εικόνα που περιέχει κείμενο, στιγμιότυπο οθόνης

Περιγραφή που δημιουργήθηκε αυτόματα

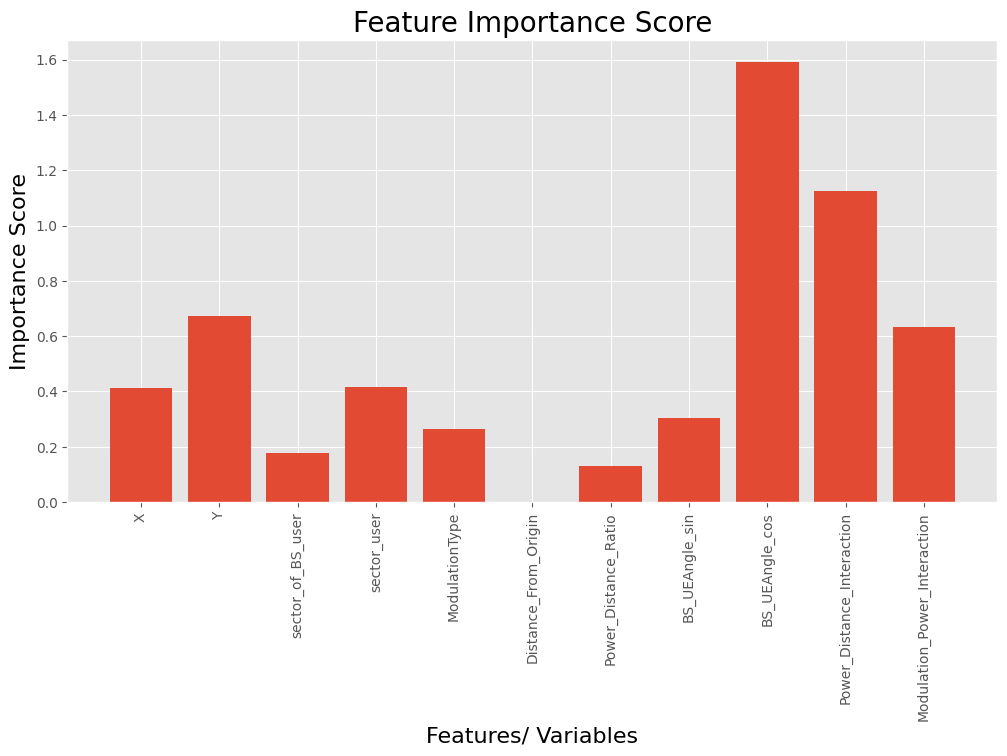
Η συσχέτιση χαρακτηριστικών μετά την αφαίρεση

1. **Κλιμάκωση Χαρακτηριστικών**

Για να εξασφαλιστεί η ομοιομορφία των δεδομένων και να αποφευχθεί η υπερβολική επιρροή χαρακτηριστικών με μεγάλες αριθμητικές τιμές, εφαρμόστηκε ο MinMaxScaler. Αυτή η μέθοδος κανονικοποίησε όλες τις αριθμητικές μεταβλητές στην κλίμακα [0, 1], διατηρώντας την σχετική πληροφορική τους αξία.

1. **Εφαρμογή Chi-Squared Feature Selection**

Η τεχνική Chi-Squared υπολογίζει τη στατιστική εξάρτηση μεταξύ κάθε ανεξάρτητου χαρακτηριστικού και της μεταβλητής στόχου (throughput). Για κάθε χαρακτηριστικό, υπολογίστηκε η τιμή του Chi-Squared, και τα αποτελέσματα παρουσιάζονται γραφικά στο παρακάτω διάγραμμα:



Όπως φαίνεται στο διάγραμμα, τα χαρακτηριστικά με τις υψηλότερες τιμές **Chi-Squared**, όπως το **BS\_UEAngle\_cos** και το **Power\_Distance\_Interaction**, εμφανίζουν την ισχυρότερη συσχέτιση με τη μεταβλητή στόχο. Αντιθέτως, χαρακτηριστικά με χαμηλές τιμές, όπως το **Distance\_From\_Origin** κρίθηκαν μη σημαντικά και εξαιρέθηκαν από το τελικό σύνολο.

**Αξιολόγηση Απόδοσης ML Αλγορίθμων και Ανάλυση Εκπαίδευσης-Επαλήθευσης**

**Εκπαίδευση και Αξιολόγηση ML Αλγορίθμων**

Το σύνολο δεδομένων χωρίστηκε αρχικά σε σύνολο εκπαίδευσης και επαλήθευσης με αναλογία **80%-20%**, χρησιμοποιώντας τη συνάρτηση train\_test\_split από τη βιβλιοθήκη scikit-learn. Για την εκπαίδευση χρησιμοποιήθηκαν οι εξής αλγόριθμοι με τις προεπιλεγμένες παραμέτρους τους, για το πρόβλημα classification:

* Logistic Regression
* K-Nearest neighbors (knn)
* Support Vector Machine (SVM)

Ενώ για το πρόβλημα regression χρησιμοποιήθηκαν οι:

* Support Vector Machine (SVM)
* Linear Regression

Για την αξιολόγηση, επιλέχθηκαν οι παρακάτω μετρικές απόδοσης (ML KPIs), οι πρώτες 4 για το πρόβλημα classification και οι τελευταιες για regression:

* **Precision**: Ποσοστό θετικών προβλέψεων που ήταν σωστές.
* **Recall**: Ποσοστό θετικών περιπτώσεων που αναγνωρίστηκαν σωστά.
* **F1-score**: Ο αρμονικός μέσος του precision και recall.
* **MSE** και **RMSE**: Μέση τετραγωνική απόκλιση και ρίζα της μέσης τετραγωνικής απόκλισης, για την ποσοτική αποτίμηση των σφαλμάτων.

Η απόδοση των αλγορίθμων παρουσιάζεται συνοπτικά στον ακόλουθο πίνακα:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Αλγοριθμοσ | Precision | Recall | F1-score | MSE | RMSE |
| Logistic Regression | 0.33 | 0.33 | 0.33 | - | - |
| κνν | 0.34 | 0.35 | 0.34 | - | - |
| svm | 0.32 | 0.32 | 0.32 | 403 | 349 |
| Linear regression | - | - | - | 403 | 349 |

**Σύγκριση με Θεωρία του Κεφαλαίου 3.2**

**Σχόλια για το Καλύτερο Μοντέλο**

* **SVM**:
  + Η απόδοσή της παραμένει σχετικά καλή, αλλά με αυξημένες λανθασμένες προβλέψεις στις τάξεις, ιδιαίτερα στην Κλάση 2.
  + Οι σωστές προβλέψεις βρίσκονται στις τιμές: 158, 246, και 248.
* **kNN**:
  + Βελτιώνει την ακρίβεια στις Κλάσεις 0 και 1 σε σχέση με την προηγούμενη μέτρηση, με τις σωστές προβλέψεις να είναι: 310, 248, και 138.
  + Παρόλα αυτά, η Κλάση 2 παραμένει πιο προβληματική με υψηλότερη διάσπαρση λανθασμένων προβλέψεων.
* **Λογιστική Παλινδρόμηση**:
  + Εμφανίζει υψηλή λανθασμένη πρόβλεψη σε όλες τις τάξεις, με σωστές προβλέψεις: 138, 232, και 303.
  + Φαίνεται να μην αποδίδει ικανοποιητικά σε αυτό το πρόβλημα ταξινόμησης.

Η σύγκριση των δύο μοντέλων regression αποκαλύπτει ότι το SVR έχει οριακά καλύτερη απόδοση, αλλά η διαφορά είναι αμελητέα:

1. Διαφορές Απόδοσης:

* Το SVR έχει ελαφρώς χαμηλότερο MSE (διαφορά περίπου 76 μονάδων)
* Η διαφορά στο RMSE είναι μόλις 0,09 μονάδες
* Το MAE διαφέρει κατά 0,005 μονάδες
* Η διαφορά στο R² είναι 0,0005 μονάδες

1. Κρίσιμες Παρατηρήσεις:

* Και τα δύο μοντέλα έχουν αρνητικό R², που υποδηλώνει ότι αποδίδουν χειρότερα από έναν απλό μέσο όρο
* Η εξηγούμενη διακύμανση είναι αρνητική και για τα δύο μοντέλα
* Τα σφάλματα είναι εξαιρετικά υψηλά σε σχέση με τα δεδομένα

**Θεωρητική Αντιστοιχία με το Κεφάλαιο 3.2**

* Η **SVM** εξακολουθεί να θεωρείται κατάλληλη για προβλήματα με γραμμικό ή ημιγραμμικό διαχωρισμό, αλλά τα νέα δεδομένα δείχνουν πιθανές δυσκολίες στον χειρισμό πιο σύνθετων μοτίβων.
* Το **kNN** επηρεάζεται από την επιλογή του αριθμού γειτόνων και είναι πιο ευαίσθητο σε μη γραμμικά όρια απόφασης.
* Η **Λογιστική Παλινδρόμηση**, παρότι απλή, δεν φαίνεται να διαχειρίζεται καλά τη σύνθετη δομή δεδομένων του προβλήματος.

**Συμπεράσματα**

* Από την ανάλυση των νέων πινάκων σύγχυσης, το μοντέλο **kNN** φαίνεται να αποδίδει καλύτερα στις Κλάσεις 0 και 1, αλλά παραμένει προβληματικό για την Κλάση 2.
* Η **SVM**, παρά τις δυσκολίες, διατηρεί μια ισορροπία απόδοσης.
* Η **Λογιστική Παλινδρόμηση** υπολείπεται σημαντικά και δεν φαίνεται κατάλληλη για αυτό το πρόβλημα.

Στο τέλος, επιπλέον πραγματοποιήσαμε hyperparameter tuning με τη χρήση του Optuna, αλλά τελικά τα αποτελέσματα ήταν παρόμοια.

**Bonus Ερωτήματα**

Για μια πιο αποτελεσματική εύρεση των υπερπαραμέτρων που βελτιστοποιούν την απόδοση των ML αλγορίθμων, γίνεται χρήση StratifiedKFold 10 μαζί με gridSearch όπου οι παράμετροι που χρησιμοποιούμε είναι:

**"SVC"**:   
 **'kernel'**: [**'linear'**, **'rbf'**, **'poly'**],  
 **'C'**: [0.1, 1, 10, 100],  
 **'gamma'**: [**'scale'**, 0.001, 0.01, 0.1, 1],  
 **'class\_weight'**: [**None**, **'balanced'**]  
,  
**"KNN"**:   
 **'n\_neighbors'**: list(range(1, 51)),  
 **'weights'**: [**'uniform'**, **'distance'**],  
 **'metric'**: [**'euclidean'**, **'manhattan'**, **'minkowski'**, **'chebyshev'**],  
 **'p'**: [1, 2, 3, 4, 5]  
,  
**"LogisticRegression"**: [  
   
 **'penalty'**: [**'l1'**],  
 **'solver'**: [**'saga'**],  
 **'C'**: [0.1, 1, 10, 100],  
 **'l1\_ratio'**: [0.0, 0.25, 0.5, 0.75, 1.0]  
 ,  
   
 **'penalty'**: [**'l2'**],  
 **'solver'**: [**'lbfgs'**, **'sag'**, **'saga'**, **'newton-cg'**],  
 **'C'**: [0.1, 1, 10, 100]  
 },  
 {  
 **'penalty'**: [**'elasticnet'**],  
 **'solver'**: [**'saga'**],  
 **'C'**: [0.1, 1, 10, 100],  
 **'l1\_ratio'**: [0.0, 0.25, 0.5, 0.75, 1.0]  
 ,  
   
 **'penalty'**: [**None**],  
 **'solver'**: [**'lbfgs'**, **'sag'**, **'saga'**, **'newton-cg'**],  
 **'C'**: [0.1, 1, 10, 100]  *Παρόλο που 'C' δεν χρησιμοποιείται με penalty='none', το περιλαμβάνουμε για συνέπεια*

*,****"LinearRegression"****:* ***'fit\_intercept'****: [****True****,* ***False****],*

Από τις παραπάνω παραμέτρους οι βέλτιστες ήταν οι :

SVC:

C: 10

class\_weight: balanced

gamma: scale

kernel: poly

KNN:

metric: manhattan

n\_neighbors: 14

p: 1

weights: uniform

LogisticRegression:

C: 100

penalty: l2

solver: sag

LinearRegression:

fit\_intercept: True

Και η απόδοση των βελτιστοποιημένων αλγορίθμων παρουσιάζεται συνοπτικά στον ακόλουθο πίνακα:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Αλγόριθμος | Precision | Recall | F1-score | MSE | RMSE |
| Logistic Regression | 0.35 | 0.35 | 0.33 | - | - |
| KNN | 0.47 | 0.47 | 0.47 | - | - |
| SVM | 0.4 | 0.4 | 0.4 | 162 | 403 |
| Linear regression | - | - | - | 162 | 403 |

Όπως είναι εμφανές από τον παραπάνω πίνακα, υπάρχει σημαντική βελτίωση της απόδοσης των μοντέλων μετά την εφαρμογή StratifiedKFold 10 μαζί με gridSearch, ιδιαίτερα στον αλγόριθμο ΚΝΝ όπου όπως βλέπουμε έχει με διαφορά την καλύτερη απόδοση πλέον.