ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΠΜΣ «ΕΦΑΡΜΟΣΜΕΝΗ ΣΤΑΤΙΣΤΙΚΗ»

Υποχρεωτική Εργασία στο Μάθημα: «**Στατιστικές Μέθοδοι Εξόρυξης Δεδομένων**»

Ακαδημαϊκό έτος 2024-25

Θρασύβουλος Μάστορας

ΜΕΣ24002

mes24002@unipi.gr

Πειραιάς | Φεβρουάριος 2025

Εισαγωγή

Στην παρούσα εργασία χρησιμοποιήθηκε το Adult Income Dataset από το UCI ML Repository. Το dataset φορτώθηκε χρησιμοποιώντας τη βιβλιοθήκη `ucimlrepo`, απ' όπου έγινε η ανάκτηση των δεδομένων και η μετατροπή τους σε `pandas` DataFrame. Στη συνέχεια, τα χαρακτηριστικά και η μεταβλητή-στόχος διαχωρίστηκαν σε δύο πίνακες (X και y) και το πλήρες dataset αποθηκεύτηκε σε ένα αρχείο CSV για την υλοποίηση των επιμέρους εργασιών. Σαν περιβάλλον υλοποίησης επιλέξαμε το Google Colab και τα αρχεία πηγαίου κώδικα είναι της μορφής ".ipynb".

1η Εργασία: Προετοιμασία δεδομένων & στατιστική ανάλυση

α) Καθαρισμός Δεδομένων

Αρχικά, αφού φορτώσαμε το dataset από το CSV αρχείο σε ένα DataFrame, πραγματοποιήσαμε έλεγχο για ελλιπείς τιμές μέσω της μεθόδου `isnull()`. Διαπιστώθηκαν ελλείψεις στις στήλες *workclass*, *occupation* και *native-country*. Οι τιμές αυτές επιλέξαμε να αντικατασταθούν με την κατηγορία "Unknown", καθώς δεν ήταν δυνατή η ακριβής ανάκτησή τους ούτε μια τεκμηριωμένη υπόθεση για την αντικατάστασή τους. Πέραν των ελλιπών τιμών, στις ίδιες στήλες, υπήρχε η επιπλέον τιμή "?", η οποία θεωρήθηκε επίσης ως ελλιπής και συνεπώς αντικαταστάθηκε και αυτή με "Unknown", έτσι ώστε να διατηρηθεί η συνοχή των δεδομένων.

Επιπλέον, εντοπίστηκαν ασυνέπειες στη στήλη «στόχο» *income*, όπου εμφανίζονταν τέσσερις διαφορετικές παραλλαγές των κατηγοριών "<=50K" και ">50K" αντί για δύο. Οι τιμές αυτές διορθώθηκαν ώστε να παραμείνουν μόνο οι δύο έγκυρες κατηγορίες. Τέλος, εντοπίσαμε μία λογική ασυνέπεια στις εγγραφές με workclass = "Never-worked", όπου η αντίστοιχη τιμή της στήλης occupation άλλαξε από "Unknown" σε "No-occupation" για μεγαλύτερη ακρίβεια αφού όσοι δεν έχουν δουλέψει ποτέ δεν χαρακτηρίζονται από την άσκηση κάποιου επαγγέλματος.

β) Μετασχηματισμός Δεδομένων

Για τη βελτίωση της ποιότητας των δεδομένων, εφαρμόστηκαν οι παρακάτω μετασχηματισμοί:

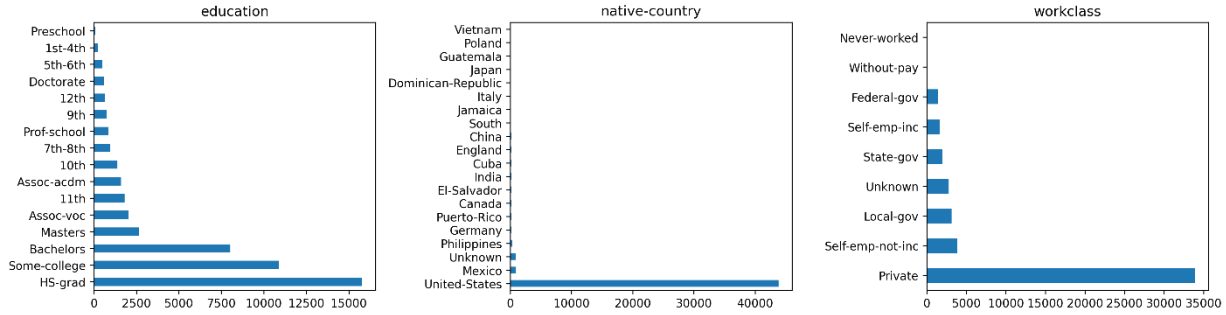
- Η στήλη *age* διακριτοποιήθηκε σε τρεις ομάδες: "Young" για ηλικίες από 17 έως 34 ετών, "Middle-aged" για ηλικίες από 35 έως 64 ετών και "Senior" για ηλικίες από 65 ετών και άνω, χρησιμοποιώντας τη συνάρτηση `pd.cut()` της `pandas`.
- Αφού πραγματοποιήσαμε γραφική αναπαράσταση των τιμών όλων των στηλών του dataset, παρατηρήσαμε πως ορισμένες στήλες περιείχαν πολλές κατηγορίες, οι οποίες μάλιστα δεν αντιπροσωπεύονταν επαρκώς από την πλειοψηφία των εγγραφών. Για τον λόγο αυτό, συγκεκριμένα στις στήλες *education*, *native-country* και *workclass* ενοποιήσαμε ορισμένες κατηγορίες, προκειμένου να μειωθεί ο αριθμός των κατηγοριών και να βελτιωθεί η ανάλυση των δεδομένων και η επίδοση των μεθόδων εξόρυξης.
    - Στη στήλη *education*, τα εκπαιδευτικά επίπεδα ομαδοποιήθηκαν ως εξής:
        - "Preschool", "1st-4th", ""5th-6th", "7th-8th" → "No-Formal-Education"
        - "9th", "10th", "11th", "12th" → "Some-High-School"
        - "HS-grad" → "High-School-Graduate"

- “Some-college”, “Assoc-voc”, “Assoc-acdm” → “Some-College”
- “Bachelors” → “Bachelors-Degree”
- “Masters”, “Prof-school”, “Doctorate” → “Advanced-Degree”

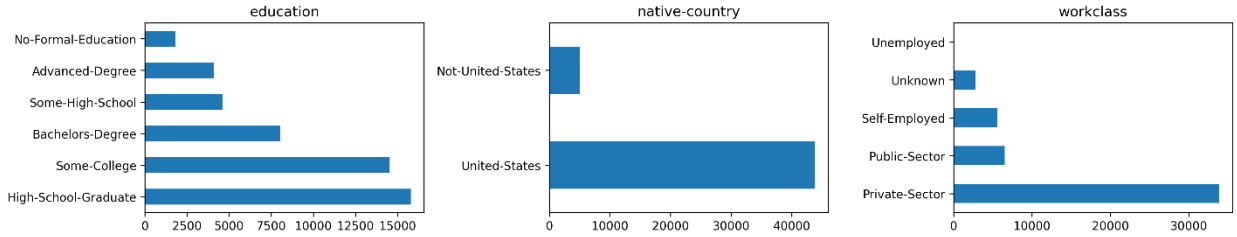
○ Στη στήλη *native-country*, έγινε ενοποίηση των χωρών σε δύο βασικές κατηγορίες: “United-States” και “Not-United-States”. Η απόφαση αυτή βασίστηκε στο γεγονός ότι η πλειονότητα των εγγραφών (σχεδόν το 90%) αφορούσαν τις ΗΠΑ, ενώ όλες οι υπόλοιπες χώρες είχαν πολύ μικρό ποσοστό εμφάνισης.

○ Στη στήλη *workclass*, οι διαφορετικές μορφές απασχόλησης ομαδοποιήθηκαν ως εξής:

- “Private” → “Private-Sector”, για όσους εργάζονται στον ιδιωτικό τομέα.
- “Local-gov”, “State-gov”, “Federal-gov” → “Public-Sector”, για κρατικούς υπαλλήλους.
- “Self-emp-not-inc”, “Self-emp-inc” → “Self-Employed”, για ελεύθερους επαγγελματίες.
- “Without-pay”, “Never-worked” → “Unemployed”, για ανέργους ή όσους δεν εργάστηκαν ποτέ.
- Η κατηγορία “Unknown” παρέμεινε ως είχε.



Εικόνα 1: Κατανομή στηλών education, native-country και workclass πριν από την ενοποίηση κατηγοριών.



Εικόνα 2: Κατανομή στηλών education, native-country και workclass μετά από την ενοποίηση κατηγοριών.

- Όλα τα αριθμητικά χαρακτηριστικά κανονικοποιήθηκαν στο εύρος [0, 1], με χρήση του **MinMaxScaler** της `scikit-learn`, εξασφαλίζοντας μια ομοιόμορφη κλίμακα και αποτρέποντας τη διαφορά τάξεων μεγέθους μεταξύ των χαρακτηριστικών.
- Όλα τα κατηγορικά χαρακτηριστικά κωδικοποιήθηκαν σε αριθμητικές τιμές μέσω **Label Encoding**, μετατρέποντάς τες σε μορφή κατάλληλη για χρήση σε τεχνικές μηχανικής μάθησης.

γ) Μείωση όγκου δεδομένων

Για τη μείωση του αριθμού χαρακτηριστικών, εφαρμόστηκε η μέθοδος **PCA** με στόχο τη διατήρηση του 95% της διακύμανσης των δεδομένων. Η επιλογή αυτή έγινε ορίζοντας την παράμετρο `n_components=0.95` στη συνάρτηση `PCA()` της `scikit-learn`, διασφαλίζοντας ότι οι πρώτες κύριες συνιστώσες διατηρούν το 95% της συνολικής πληροφορίας του dataset. Η PCA μείωσε τον αριθμό των χαρακτηριστικών από 14 σε 6, διατηρώντας τη μέγιστη δυνατή πληροφορία.

Όσον αφορά τη μείωση του αριθμού εγγραφών, εξετάστηκαν δύο τεχνικές δειγματοληψίας:

- **Απλή τυχαία δειγματοληψία**, όπου επιλέχθηκε τυχαία το 50% των εγγραφών, ανεξάρτητα από την κατηγορία της στήλης-στόχου *income*.
- **Διαστρωματωμένη δειγματοληψία**, ώστε να διατηρηθεί η αρχική κατανομή της στήλης income και να μειωθεί ο κίνδυνος εισαγωγής μεροληψίας. Η επιλογή των δειγμάτων έγινε ξεχωριστά για κάθε κατηγορία εισοδήματος, χρησιμοποιώντας τη μέθοδο `groupby("income")`, η οποία διαχωρίζει τα δεδομένα σε ομάδες με βάση την τιμή της στήλης *income*. Στη συνέχεια, από κάθε ομάδα έγινε τυχαία δειγματοληψία του 50% των εγγραφών μέσω της μεθόδου `sample(frac=0.5)`. Έτσι, διασφαλίστηκε ότι οι αναλογίες στο δείγμα παρέμειναν ίδιες με το αρχικό dataset, ενώ η επιλογή των δειγμάτων εντός κάθε ομάδας παρέμεινε τυχαία.

2η Εργασία: Επιλογή Χαρακτηριστικών & Ανίχνευση Ακραίων Τιμών

α) Επιλογή Χαρακτηριστικών (Feature Selection)

Χρησιμοποιώντας το προεπεξεργασμένο dataset που προέκυψε από την 1η Εργασία, εφαρμόστηκε η μέθοδος **SelectKBest** από τη βιβλιοθήκη `scikit-learn` για την επιλογή των k πιο σημαντικών χαρακτηριστικών. Ως κριτήριο επιλογής χρησιμοποιήθηκε το **chi2** (`score_func=chi2`), το οποίο μετρά την ανεξαρτησία μεταξύ κάθε χαρακτηριστικού και της μεταβλητής-στόχου income. Χαρακτηριστικά με υψηλότερη βαθμολογία θεωρούνται περισσότερο σχετιζόμενα με την πρόβλεψη της μεταβλητής-στόχου. Έπειτα από δοκιμές για διάφορες τιμές της παραμέτρου k, διαπιστώθηκε ότι η τιμή 7 απέδιδε τη μεγαλύτερη ακρίβεια (accuracy). Τα 7 αυτά χαρακτηριστικά ήταν τα εξής:

- age
- education
- marital-status
- relationship
- sex
- capital-gain
- capital-loss

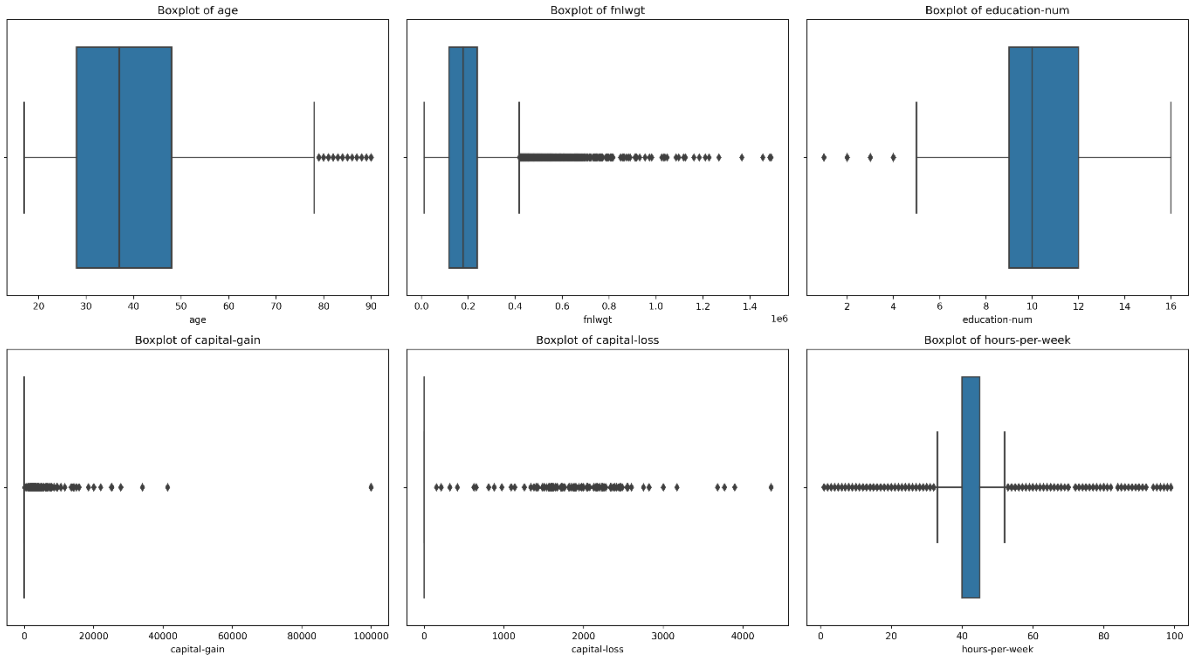
Μετά την επιλογή των χαρακτηριστικών, πραγματοποιήθηκε εκπαίδευση ενός ταξινομητή δέντρου απόφασης (**DecisionTreeClassifier**), τόσο με όλα τα χαρακτηριστικά (14 συνολικά) όσο και με τα 7 σημαντικότερα. Ο χρόνος εκπαίδευσης μετρήθηκε μέσω της

βιβλιοθήκης `time`, πριν και μετά την εκπαίδευση του ταξινομητή. Η ανάλυση των αποτελεσμάτων έδειξε ότι η ακρίβεια του μοντέλου αυξήθηκε από 81% σε 86%, ενώ ο χρόνος εκπαίδευσης μειώθηκε κατά περίπου 80%. Συνεπώς, η επιλογή των πιο σημαντικών χαρακτηριστικών συνέβαλε στη βελτίωση της απόδοσης του ταξινομητή, διατηρώντας υψηλή ακρίβεια και μειώνοντας παράλληλα το υπολογιστικό κόστος.

β) Ανίχνευση Ακραίων Τιμών (Outlier Detection)

Η διαδικασία της ανίχνευσης των ακραίων τιμών επιλέξαμε να γίνει στο αρχικό dataset, πριν από το στάδιο της προεπεξεργασίας, έτσι ώστε τα αριθμητικά χαρακτηριστικά να διατηρήσουν τις αρχικές τους τιμές και να μην επηρεαστούν από την κανονικοποίηση στο διάστημα [0, 1].

Αρχικά, έγινε οπτικοποίηση των αριθμητικών μεταβλητών μέσω boxplots για τον οπτικό εντοπισμό πιθανών ακραίων τιμών. Συγκεκριμένα, για τις μεταβλητές *age*, *fnlwgt*, *education-num*, *capital-gain*, *capital-loss* και *hours-per-week* δημιουργήθηκαν ξεχωριστά boxplots που να απεικονίζουν την κατανομή των δεδομένων και τις πιθανές αποκλίσεις από τα συνήθη όρια.



Εικόνα 1: Boxplots για την οπτικοποίηση των ακραίων τιμών σε κάθε αριθμητικό χαρακτηριστικό.

Για την ανίχνευση των ακραίων τιμών, χρησιμοποιήθηκε η τεχνική **Interquartile Range (IQR)**. Υπολογίστηκαν τα τεταρτημόρια Q1 (25ο εκατοστημόριο) και Q3 (75ο εκατοστημόριο), με το εύρος IQR να προκύπτει από τη διαφορά μεταξύ αυτών των δύο τιμών. Οι ακραίες τιμές ορίστηκαν ως εκείνες που βρίσκονται εκτός των ορίων:

$$Lower\ bound = Q1 - 1.5 \cdot IQR$$
$$Upper\ bound = Q3 + 1.5 \cdot IQR$$

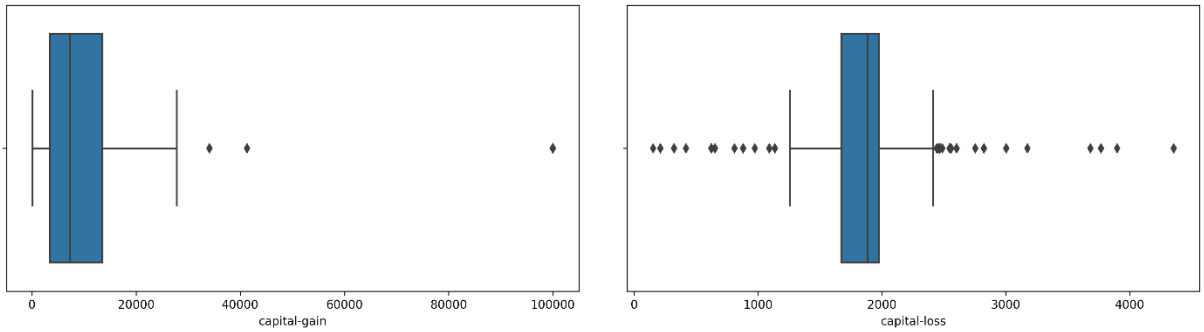
Η διαδικασία αυτή υλοποιήθηκε μέσω της συνάρτησης `detect_outliers_iqr()`, η οποία δέχεται ως όρισμα μια στήλη (`column`) του dataset και επιστρέφει τις εγγραφές στις

οποίες εντοπίζονται ακραίες τιμές στη συγκεκριμένη column, όπως επίσης και τα αντίστοιχα άνω και κάτω όρια, όπως υπολογίζονται από την τεχνική IQR.

Όσον αφορά τη μεταβλητή *age*, δεδομένου ότι έχει ήδη διακριτοποιηθεί σε τρεις ομάδες (βλ. 1η Εργασία), δεν κρίθηκε σκόπιμη η τροποποίηση ή αφαίρεση των ακραίων τιμών της. Εξάλλου, οι υψηλές τιμές που εντοπίστηκαν αφορούσαν κυρίως άτομα άνω των 80 ετών, τα οποία θεωρούνται αποδεκτές περιπτώσεις. Αντίστοιχα, στη μεταβλητή *education-num*, οι ακραίες τιμές αντιστοιχούσαν σε άτομα με πολύ λίγα χρόνια εκπαίδευσης, γεγονός που, αν και σπάνιο στο dataset, αποτελεί υπαρκτό σενάριο.

Αντίθετα, στις μεταβλητές *fnlwgt* και *hour-per-week* εφαρμόστηκε η μέθοδος **winsorization**, κατά την οποία οι ακραίες τιμές αυτών των δύο στηλών αντικαθίστανται με τα όρια των αποδεκτών τιμών, ώστε να μειωθεί η επίδρασή τους χωρίς να αφαιρεθούν δεδομένα.

Για τις μεταβλητές *capital-gain* και *capital-loss*, επειδή η πλειοψηφία των εγγραφών περιείχαν μηδενικές τιμές, με αποτέλεσμα αυτό να δυσκολεύει τον εντοπισμό των πραγματικά ακραίων τιμών, αποφασίσαμε να απομονώσουμε τις μη μηδενικές τιμές των δύο αυτών στηλών και σ' αυτές να εφαρμόσουμε τη τεχνική IQR. Οι ακραίες τιμές που εντοπίστηκαν αντιμετωπίστηκαν με winsorization, διατηρώντας τις τιμές εντός των επιτρεπτών ορίων που είχαν προσδιοριστεί μέσω του IQR.



Εικόνα 2: Boxplots για την οπτικοποίηση των ακραίων τιμών στις μη μηδενικές τιμές των capital-gain και capital-loss.

Συνολικά, η προσέγγιση που ακολουθήσαμε για τη διαχείριση των ακραίων τιμών απέφυγε την απομάκρυνση εγγραφών με ακραίες τιμές, προκειμένου να διατηρηθεί η πληροφορία που θα μπορούσε να είναι χρήσιμη στις επόμενες διαδικασίες συσταδοποίησης και κατηγοριοποίησης.

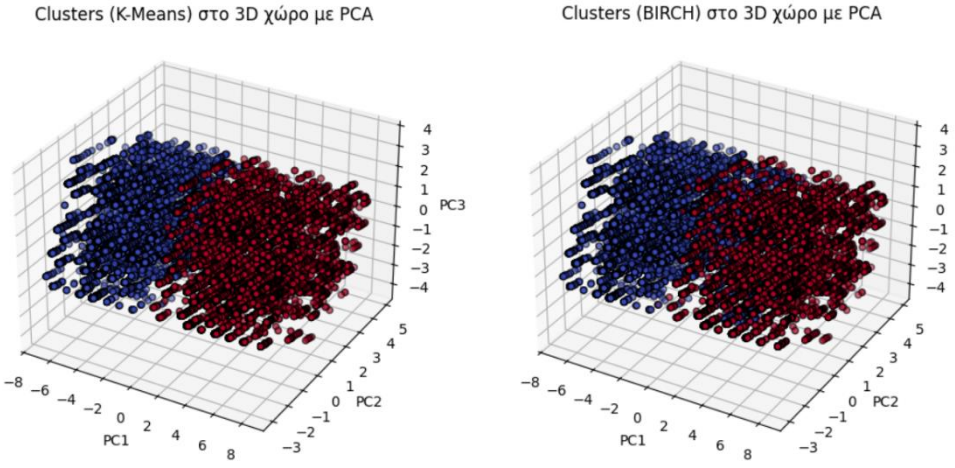
3η Εργασία: Συσταδοποίηση (Clustering)

Στα πλαίσια υλοποίησης της παρούσας εργασίας, η μεταβλητή-στόχος *income* παραλείφθηκε από τη διαδικασία εκπαίδευσης των αλγορίθμων συσταδοποίησης, αλλά αξιοποιήθηκε εκ των υστέρων ως σημείο αναφοράς (ground truth) για την αξιολόγηση των αποτελεσμάτων. Η ανάλυση πραγματοποιήθηκε με δύο διαφορετικές τεχνικές συσταδοποίησης: τον αλγόριθμο **K-Means** και τον **BIRCH**, συγκρίνοντας την απόδοσή τους.

Αρχικά, ο K-Means εφαρμόστηκε με δύο συστάδες (`n_clusters=2`), επιχειρώντας να διαχωρίσει τις εγγραφές σε δύο ομάδες που πιθανώς ανταποκρίνονται στις δύο κατηγορίες εισοδήματος ("<=50K" και ">50K"). Τα αποτελέσματα συγκρίθηκαν με τις πραγματικές τιμές της μεταβλητής income, ενώ η οπτικοποίησή τους έγινε μέσω γραφημάτων κατανομής και

αναπαράστασης σε δύο διαστάσεις χρησιμοποιώντας την τεχνική PCA, ώστε να γίνει κατανοητή η διαμόρφωση των συστάδων σε σχέση με τα αρχικά δεδομένα των 14 χαρακτηριστικών.

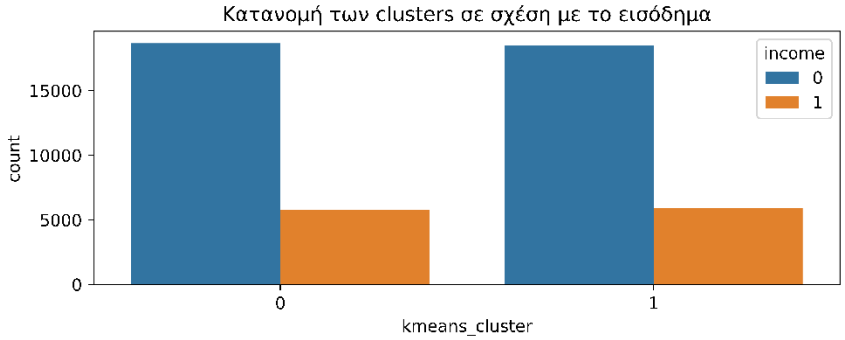
Αντίστοιχα, ο BIRCH, ο οποίος εφαρμόστηκε με δύο συστάδες και κατάλληλη ρύθμιση της παραμέτρου κατωφλίου (`threshold`). Τα αποτελέσματά του αναλύθηκαν και οπτικοποιήθηκαν με τον ίδιο τρόπο. Επιπλέον, για πληρέστερη κατανόηση της συμπεριφοράς των αλγορίθμων, τα αποτελέσματα προβλήθηκαν και σε τρεις διαστάσεις, μέσω PCA και τρισδιάστατων διαγραμμάτων.

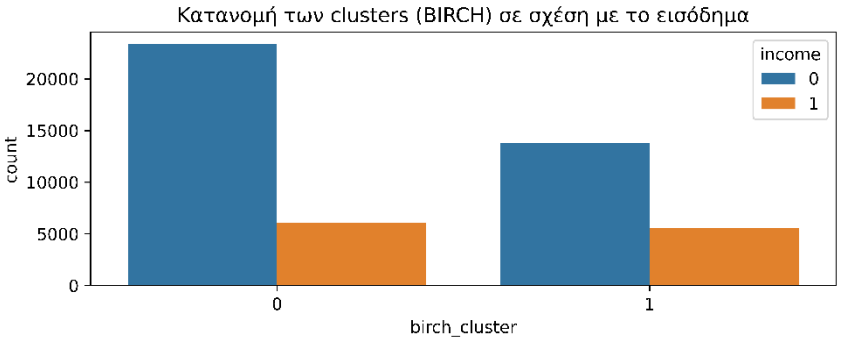


Εικόνα 3: Απεικόνιση clusters με τρισδιάστατα διαγράμματα.

Για την ποσοτική αξιολόγηση των αποτελεσμάτων, χρησιμοποιήθηκε ο δείκτης **Adjusted Rand Index (ARI)**, ο οποίος μετράει την ομοιότητα μεταξύ των συστάδων που προέκυψαν από τον αλγόριθμο και των πραγματικών κατηγοριών του εισοδήματος (`y_true`). Οι τιμές που προέκυψαν ήταν $3.93 \cdot 10^{-5}$ για τον K-Means και 0.023 για τον BIRCH. Αυτές οι χαμηλές τιμές υποδηλώνουν ότι οι συστάδες που δημιουργήθηκαν δεν ανταποκρίνονται ικανοποιητικά στη διάκριση των ατόμων με βάση το εισόδημα, επιβεβαιώνοντας ότι τα διαθέσιμα χαρακτηριστικά δεν περιέχουν επαρκή πληροφορία για έναν φυσικό διαχωρισμό μέσω μη επιβλεπόμενης μάθησης.

Όπως φαίνεται από τα παρακάτω γραφήματα κατανομής, η μεγαλύτερη πρόκληση αφορά την ορθότητα της ομαδοποίησης των δειγμάτων που ανήκουν στην κατηγορία εισοδήματος “>50K”, η οποία αποτελεί μειοψηφία στο dataset (περίπου 20% έναντι 80% της κατηγορίας “<=50K”). Στην περίπτωση του K-Means, η κατανομή των συστάδων είναι σχεδόν ισομερής, χωρίς σαφή διάκριση μεταξύ των κατηγοριών εισοδήματος. Αντίθετα, ο BIRCH μειώνει τον αριθμό των εγγραφών της κατηγορίας “<=50K” που κατατάσσονται λανθασμένα στην ομάδα του υψηλότερου εισοδήματος, βελτιώνοντας ελαφρώς τη διαφοροποίηση μεταξύ των δύο συστάδων.





Εικόνα 4: Γραφήματα κατανομής των label clusters σε σύγκριση με τα ground truth labels.

4η Εργασία: Κατηγοριοποίηση (Classification) – Υλοποίηση με scikit-learn

Το dataset χωρίστηκε σε σύνολο εκπαίδευσης (80%) και σύνολο ελέγχου (20%), χρησιμοποιώντας τη μέθοδο `train_test_split()`, γεγονός που επιτυγχάνεται θέτοντας την παράμετρο `test_size=0.2`. Ο πρώτος ταξινομητής που χρησιμοποιήθηκε ήταν ο **Random Forest**, ένας ισχυρός αλγόριθμος που συνδυάζει πολλά δέντρα απόφασης για τη λήψη πιο αξιόπιστων προβλέψεων. Η αξιολόγηση του ταξινομητή πραγματοποιήθηκε με τρεις βασικούς στατιστικούς δείκτες: **Accuracy**, **Precision** και **Recall**. Το accuracy μετράει το ποσοστό των σωστά ταξινομημένων περιπτώσεων στο σύνολο δεδομένων, ενώ το precision δείχνει το ποσοστό των περιπτώσεων που προβλέφθηκαν ως θετικές και ήταν όντως σωστές. Τέλος, το recall αποτιμά την ικανότητα του ταξινομητή να εντοπίζει τις θετικές περιπτώσεις, μετρώντας πόσα από τα πραγματικά θετικά δείγματα αναγνωρίστηκαν σωστά.

Παράλληλα, και για την οπτική ανάλυση των σφαλμάτων ταξινόμησης, δημιουργήθηκε και πίνακας σύγχυσης (**confusion matrix**), ο οποίος αποτυπώνει τον αριθμό των σωστών και λανθασμένων προβλέψεων σε κάθε κατηγορία. Ο πίνακας αυτός επιτρέπει την καλύτερη κατανόηση της συμπεριφοράς του ταξηνομητή, επισημαίνοντας περιπτώσεις λανθασμένων προβλέψεων είτε αυτές αφορούν «ψευδώς θετικές» είτε «ψευδώς αρνητικές».

Για τη βελτίωση της απόδοσης του ταξινομητή, εφαρμόστηκε η τεχνική **GridSearchCV**, η οποία η οποία δοκιμάζει διάφορους συνδυασμούς παραμέτρων για να εντοπίσει τις πιο αποδοτικές ρυθμίσεις. Σαν είσοδο λαμβάνει ένα λεξικό παραμέτρων (`param_grid`), όπου καθορίζονται οι τιμές που θα δοκιμαστούν για κάθε παράμετρο. Στην περίπτωση του Random Forest, εξετάστηκαν διαφορετικές τιμές για τον αριθμό των δέντρων (`n_estimators`), το μέγιστο βάθος τους (`max_depth`) και τον ελάχιστο αριθμό δειγμάτων που απαιτείται σε κάθε φύλλο του δέντρου (`min_samples_leaf`). Η τεχνική αυτή εκπαίδευσε επανειλημμένα τον ταξινομητή με όλους τους δυνατούς συνδυασμούς αυτών των τιμών και επιλέχθηκε εκείνος που παρείχε το μέγιστο accuracy, το οποίο ανήλθε σε 87%.

Στη συνέχεια, δοκιμάστηκε και ένας δεύτερος ταξινομητής, ο **K-Nearest Neighbors (KNN)**. Αρχικά, εκπαιδεύτηκε με τις προεπιλεγμένες παραμέτρους, όπου ο αριθμός των γειτόνων είναι 5 (`n_neighbors=5`), ενώ στη συνέχεια χρησιμοποιήθηκε και πάλι το GridSearchCV για την εύρεση της βέλτιστης τιμής της παραμέτρου `n_neighbors`, δηλαδή του αριθμού των γειτονικών δειγμάτων που λαμβάνονται υπόψη για την ταξινόμηση. Από τη διαδικασία αυτή, προέκυψε ότι καλύτερη απόδοση επιτεύχθηκε με 40 γείτονες, οδηγώντας σε accuracy 84%.

Η σύγκριση των δύο ταξινομητών έδειξε ότι ο Random Forest παρουσίασε υψηλότερη ακρίβεια σε σχέση με τον KNN. Επιπλέον, η χρήση του GridSearchCV συνέβαλε στη βελτιστοποίηση των δύο τεχνικών ταξινόμησης, αυξάνοντας την ακρίβεια των προβλέψεων και μειώνοντας τα σφάλματα ταξινόμησης. Συνολικά, η καλύτερη απόδοση επιτεύχθηκε με τον βελτιστοποιημένο Random Forest, καθιστώντας τον πιο αποτελεσματικό ταξινομητή για το συγκεκριμένο σύνολο δεδομένων.

Τεχνική ταξινόμησης	Accuracy	Precision	Recall
Random Forest	1.00	1.00	1.00
Random Forest (`n_estimators=200, max_depth=20, min_samples_leaf=4`)	0.89	0.83	0.66
K-Nearest Neighbors	0.87	0.78	0.66
K-Nearest Neighbors (`n_neighbors=40`)	0.84	0.73	0.54

Εικόνα 5: Στατιστικοί δείκτες στο σύνολο εκπαίδευσης.

Τεχνική ταξινόμησης	Accuracy	Precision	Recall
Random Forest	0.85	0.72	0.62
Random Forest (`n_estimators=200, max_depth=20, min_samples_leaf=4`)	0.87	0.78	0.61
K-Nearest Neighbors	0.83	0.67	0.55
K-Nearest Neighbors (`n_neighbors=40`)	0.84	0.73	0.52

Εικόνα 6: Στατιστικοί δείκτες στο σύνολο ελέγχου.

5η Εργασία: Classification – Υλοποίηση με keras/tensorflow

Ξεκινώντας, ο διαχωρισμός των δεδομένων πραγματοποιήθηκε με τη μέθοδο `train_test_split()`, εξασφαλίζοντας ότι κατανομή της κατηγορίας-στόχου παραμένει ισορροπημένη μέσω της παραμέτρου `stratify=y`. Η διαδικασία έγινε σε δύο στάδια. Πρώτα, το 60% των δεδομένων διατέθηκε για εκπαίδευση (`test_size=0.4`). Στη συνέχεια, το υπόλοιπο 40% χωρίστηκε περαιτέρω ώστε το 10% να χρησιμοποιηθεί για αξιολόγηση και το 30% για έλεγχο (`test_size=0.75`, αφού το 75% του 40% αντιστοιχεί στο 30% του συνόλου).

Για τη δημιουργία του πολυεπίπεδου νευρωνικού δικτύου (MLP) χρησιμοποιήθηκε το `Sequential` API της βιβλιοθήκης `keras`, το οποίο επιτρέπει τη διαμόρφωση του μοντέλου μέσω διαδοχικής στοίβαξης επιπέδων. Η αρχιτεκτονική του περιλαμβάνει τα παρακάτω επίπεδα:

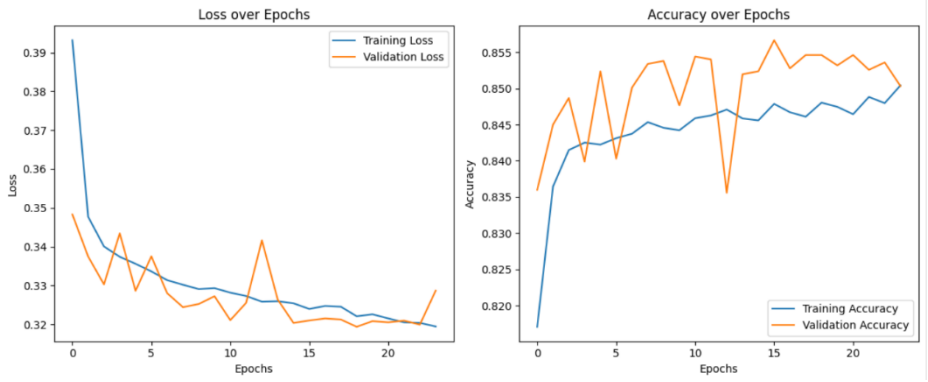
- **Στοιβάδα εισόδου**: Ένα `Dense` επίπεδο με 64 νευρώνες και συνάρτηση ενεργοποίησης ReLU. Ο αριθμός των εισόδων (`input_dim`) καθορίζεται από το πλήθος των χαρακτηριστικών του dataset (`X_train.shape[1]`).
- **Κρυφές στοιβάδες**:
  - Πρώτη κρυφή στοιβάδα με 64 νευρώνες και ReLU,
  - Δεύτερη κρυφή στοιβάδα με 32 νευρώνες και ReLU, όπου μειώνεται σταδιακά ο αριθμός των νευρώνων όσο το δίκτυο προχωρά προς την έξοδο.
- **Στοιβάδα εξόδου**: Περιλαμβάνει έναν μόνο νευρώνα με συνάρτηση ενεργοποίησης sigmoid, κατάλληλη για δυαδική ταξινόμηση. Η έξοδος αυτού του νευρώνα είναι μια τιμή μεταξύ 0 και 1, η οποία ερμηνεύεται ως η πιθανότητα το εισόδημα να ξεπερνά ή όχι τα $50.000.

Το νευρωνικό δίκτυο μεταγλωττίστηκε (μέθοδος `compile()`) με τη συνάρτηση απώλειας `binary_crossentropy`, η οποία είναι ιδανική για δυαδικά προβλήματα ταξινόμησης. Ως βελτιστοποιητής (`optimizer`) επιλέχθηκε ο Adam, ενώ ως μέτρο αξιολόγησης επιλέχθηκε το accuracy (`metrics=["accuracy"]`).

Το μοντέλο εκπαιδεύτηκε χρησιμοποιώντας τη μέθοδο `fit()`, με:

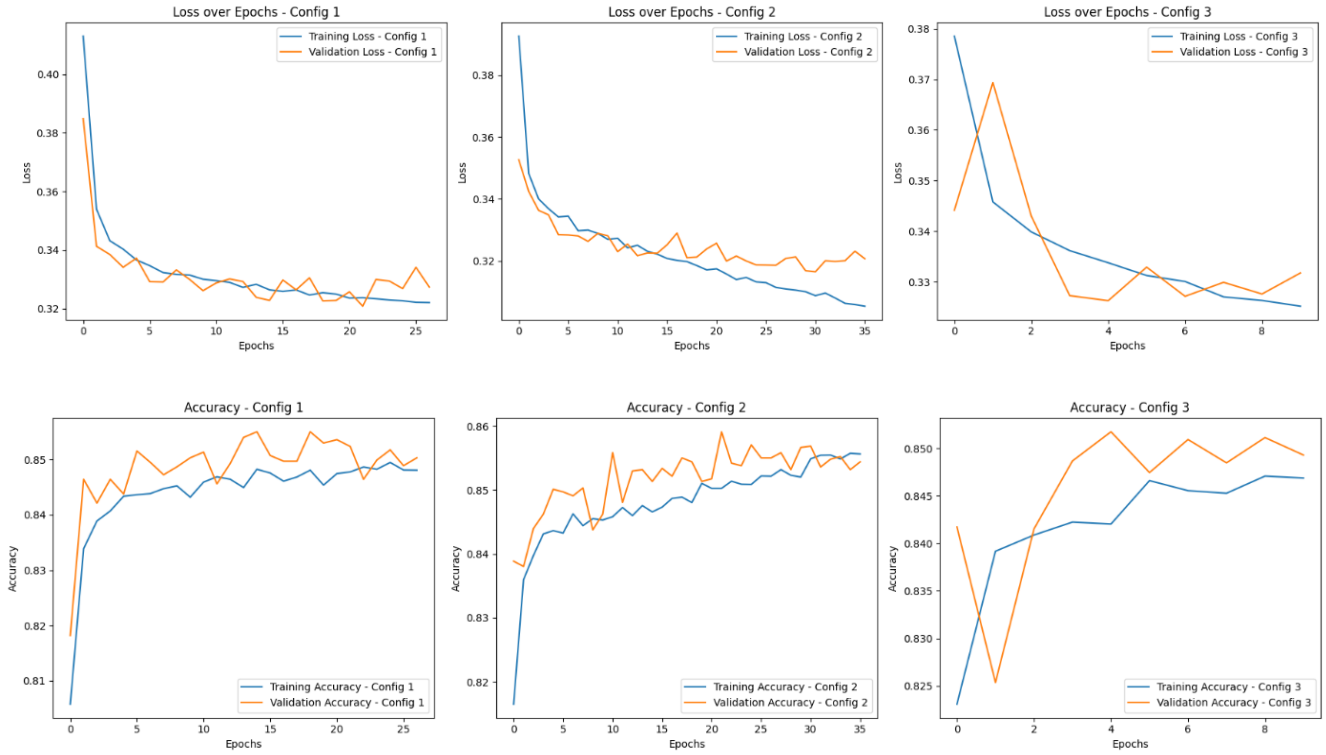
- Τα δεδομένα εκπαίδευσης και αξιολόγησης για την προσαρμογή και παρακολούθηση της απόδοσης.
- 50 εποχές με `batch_size=32`.
- Early stopping για την αποφυγή υπερπροσαρμογής, διακόπτοντας την εκπαίδευση αν η απώλεια στο σύνολο αξιολόγησης (`val_loss`) δεν βελτιωνόταν για 5 συνεχόμενες εποχές, διατηρώντας παράλληλα τα βέλτιστα βάρη (`restore_best_weights=True`).

Η απόδοση του μοντέλου αξιολογήθηκε μέσω της μεθόδου `evaluate()`, η οποία επέστρεψε δύο βασικούς δείκτες: την ακρίβεια (accuracy) και την απώλεια (loss) στο σύνολο ελέγχου, προσφέροντας μια συνολική εκτίμηση της ικανότητας γενίκευσης. Τα διαγράμματα ακρίβειας και απώλειας δείχνουν ότι τόσο η απώλεια εκπαίδευσης όσο και η απώλεια αξιολόγησης μειώνονται σταδιακά, υποδηλώνοντας ότι το μοντέλο μαθαίνει αποτελεσματικά από τα δεδομένα. Η σταθεροποίηση της απώλειας στις τελευταίες εποχές υποδεικνύει ότι η εκπαίδευση έχει συγκλίνει, ενώ η απουσία σημαντικής απόκλισης μεταξύ των δύο καμπυλών σημαίνει ότι το μοντέλο δεν παρουσιάζει έντονα σημάδια υπερπροσαρμογής (overfitting). Οι μικρές διακυμάνσεις στην απώλεια αξιολόγησης είναι αναμενόμενες και δεν επηρεάζουν σημαντικά τη συνολική συμπεριφορά του μοντέλου.



Εικόνα 7: Διαγράμματα εξέλιξης της απώλειας και της ακρίβειας ανά τις εποχές εκπαίδευσης.

Για τη βελτιστοποίηση του δικτύου, δοκιμάστηκαν τρεις διαφορετικές διαμορφώσεις νευρώνων στις κρυφές στοιβάδες: (32, 16), (128, 64) και (256, 128). Για κάθε διαμόρφωση δημιουργήθηκε νέο μοντέλο μέσω της συνάρτησης `create_NN(neurons1, neurons2)`, το οποίο εκπαιδεύτηκε για 50 εποχές με early stopping.



Εικόνα 8: Διαγράμματα εξέλιξης απώλειας και ακρίβειας για κάθε διαμόρφωση νευρώνων.

Τα αποτελέσματα έδειξαν ότι οι διαφορετικές αρχιτεκτονικές επηρεάζουν σημαντικά τόσο τη διαδικασία εκπαίδευσης όσο και την ικανότητα γενίκευσης του μοντέλου:

- Διαμόρφωση **(32, 16)**: Παρουσίασε σταθερή, αλλά ελαφρώς χαμηλότερη απόδοση, με ομαλή μείωση της απώλειας και μικρή διαφορά μεταξύ των συνόλων εκπαίδευσης και αξιολόγησης.
- Διαμόρφωση **(128, 64)**: Πέτυχε την καλύτερη ισορροπία μεταξύ εκπαίδευσης και γενίκευσης, με υψηλή ακρίβεια, σταθερή σύγκλιση και χωρίς έντονες διακυμάνσεις στην απώλεια αξιολόγησης.
- Διαμόρφωση **(256, 128)**: Αν και έμαθε ταχύτερα, εμφάνισε σημάδια υπερπροσαρμογής, με μεγαλύτερες διακυμάνσεις στην απώλεια και την ακρίβεια στο σύνολο αξιολόγησης.

Οι διαφορές αυτές οφείλονται κυρίως στον αριθμό των παραμέτρων του κάθε μοντέλου και στη σχέση του με το μέγεθος των δεδομένων εκπαίδευσης. Οι μεγαλύτερες αρχιτεκτονικές έχουν αυξημένη μαθησιακή ικανότητα, αλλά και μεγαλύτερη τάση για υπερπροσαρμογή, ενώ οι μικρότερες είναι πιο σταθερές αλλά ενδέχεται να μην αξιοποιούν πλήρως την πληροφορία του dataset. Συνεπώς, η διαμόρφωση (128, 64) αποτελεί την πιο αποτελεσματική λύση σε αυτό το πείραμα.