Ανοιχτή Άσκηση 1: Από πότε είναι αυτό το κομμάτι;

Η άσκηση αυτή δε βαθμολογείται και προσφέρεται αποκλειστικά για τη δική σας εξάσκηση.
*Όπου απαιτείται κλιμάκωση των δεδομένων χρησιμοποιήστε **pipelines**.*

Εισαγωγικά

Το Million Song Dataset (MSD) είναι ένα dataset που προορίζεται για την εκπαίδευση music recommender systems. Περιέχει μια πολύ μεγάλη συλλογή τραγουδιών, μαζί με listening histories και άλλα δεδομένα χρήσης. Τα τραγούδια δεν διατίθενται ως raw content (μεταξύ άλλων και για λόγους πνευματικών δικαιωμάτων). Αντ' αυτού, για κάθε κομμάτι παρέχονται 90 περιγραφείς του ηχοχρώματος (timbre).
Το Year Prediction MSD είναι ένα υποσύνολο του MSD που προορίζεται για ένα απλούστερο task: πρόβλεψη της χρονιάς κυκλοφορίας ενός τραγουδιού με βάση το περιεχόμενό του.

1. Εισαγωγή δεδομένων και προεργασία

Κατεβάστε το Year Prediction MSD Dataset και εισάγετέ το στο script σας. Η προβλεπόμενη πρακτική είναι να χρησιμοποιήσετε τα πρώτα 463,715 παραδείγματα για εκπαίδευση και τα επόμενα 51,630 για αξιολόγηση. Ο λόγος είναι πως είναι έτσι κατανεμημένα στο dataset ώστε να μην υπάρχουν τραγούδια του ίδιου καλλιτέχνη στο training και στο test set.
Για να μην ανεβάζετε το .csv σε κάθε νέο session, μπορείτε να το ανεβάσετε στο Google Drive σας και να συνδέσετε το Colab με το Drive. Μπορείτε μάλιστα να ανεβάσετε απευθείας το .zip στο drive, και να το κάνετε unzip μέσα από το script με τις παρακάτω εντολές για να πάρετε το csv:

```
from google.colab import drive
drive.mount('/content/gdrive')
!unzip gdrive/MyDrive/ML\ Data/YearPredictionMSD.zip
```

Όπου "`ML Data`" είναι ο φάκελος μέσα στο Drive όπου θα έχετε το .zip. Αυτές οι εντολές θα δημιουργήσουν locally μέσα στο colab το YearPredictionMSD.csv που χρειάζεστε.

Εναλλακτικά, αν θέλετε να έχετε το .csv μέσα στο drive αντί για το .zip, θα τρέξετε πάλι τη drive.mount και στη συνέχεια θα ανοίξετε το .csv με

```
open('gdrive/MyDrive/ML\ Data/YearPredictionMSD.csv')
```

Αυτό γίνεται γιατί η mount δημιουργεί ένα φάκελο gdrive/MyDrive που είναι virtual link προς το filesystem του Google Drive σας.

Αν έχετε πρόβλημα στη διαχείριση ενός τόσο μεγάλου dataset, μπορείτε να χρησιμοποιήσετε το υποσύνολο που θα βρείτε εδώ. Στην περίπτωση αυτή κάντε random train-test split με 10% test set και random_seed=42.

Τι φάσμα τιμών έχει η μεταβλητή-στόχος; Πόσες φορές εμφανίζεται η κάθε τιμή-στόχος; Τι φάσμα τιμών έχουν οι ανεξάρτητες μεταβλητές; Θα χρειαστούν κλιμάκωση κατά τη γνώμη σας;
Δεδομένου του στόχου μας (να προβλέψουμε τη χρονιά με βάση το ηχόχρωμα), θεωρείτε ότι θα είναι σωστότερο να αντιμετωπίσουμε το πρόβλημα ως παλινδρόμηση ή ταξινόμηση; Τι κάνει τη μια προσέγγιση θεωρητικά καταλληλότερη από την άλλη; (Απαντήστε με βάση το problem statement και όχι με βάση τα αποτελέσματα των επόμενων ερωτημάτων).

2. Συνάρτηση αξιολόγησης

Τα κριτήρια της ταξινόμησης (π.χ. Accuracy, F1-score) ελέγχουν κατά πόσον η χρονιά προβλέφθηκε με απόλυτη ακρίβεια ή όχι και δεν λαμβάνουν υπόψη το πόσο κοντά πέσαμε. Τα κριτήρια της

παλινδρόμησης (MSE, R^2) δε μας δίνουν κάποια διαισθητική απάντηση σχετικά με το πόσα κομμάτια εκτιμήθηκαν σωστά, παρά μόνο τη συνολική εγγύτητα της πρόβλεψης. Θα είναι ενδεχομένως βοηθητικό να δημιουργήσουμε κάποιο άλλο, δικό μας κριτήριο αξιολόγησης.
Γράψτε μία συνάρτηση η οποία θα δέχεται τα ground truths, τα predictions ενός classifier ή regressor, και έναν αριθμό N, και θα επιστρέφει το ποσοστό των predictions που απέχουν από τα ground truths κατά N χρόνια ή λιγότερο. Με ποιο κλασικό κριτήριο μοιάζει περισσότερο αυτή η ποσότητα; (π.χ. Accuracy, Precision, Recall, F1-score)

3. Ταξινόμηση

Εφαρμόστε k Nearest Neighbors classification στα δεδομένα. Δοκιμάστε grid search με 3-fold Cross Validation για 1 έως 7 γείτονες, και *Uniform* ή *Weighted* συντελεστές βαρύτητας. Στο αντικείμενο GridSearchCV δώστε *verbose=3* ώστε να παρακολουθείτε καλύτερα την εξέλιξη του search. Εμφανίστε το καλύτερο score του cross-validation, και τις αντίστοιχες καλύτερες τιμές των υπερπαραμέτρων. Εμφανίστε το F1-score στο test set, και το δικό μας κριτήριο αξιολόγησης για N=3 και N=5 χρόνια.
Εφαρμόστε Extremely Randomized Trees classification. Εμφανίστε το F1-score στο training set και στο test set, και το δικό μας κριτήριο αξιολόγησης για N=3 και N=5 χρόνια.

4. Παλινδρόμηση

Εφαρμόστε k Nearest Neighbors regression στα δεδομένα. Δοκιμάστε grid search με 3-fold Cross Validation για 1 έως 7 γείτονες, και *Uniform* ή *Weighted* συντελεστές βαρύτητας. Στο αντικείμενο GridSearchCV δώστε *verbose=3* ώστε να παρακολουθείτε καλύτερα την εξέλιξη του search. Εμφανίστε το καλύτερο score του cross-validation, και τις αντίστοιχες καλύτερες τιμές των υπερπαραμέτρων. Εμφανίστε το R^2-score στο test set, και το δικό μας κριτήριο αξιολόγησης για N=3 και N=5 χρόνια.

Εφαρμόστε Extremely Randomized Trees regression. Εμφανίστε το R^2-score στο training set και στο test set, και το δικό μας κριτήριο αξιολόγησης για N=3 και N=5 χρόνια.

5. SVMs

Θα παρατηρήσετε ότι δεν σας ζητήθηκε η χρήση SVMs για την επίλυση του προβλήματος. Γιατί συμβαίνει αυτό; Ποιο χαρακτηριστικό του dataset τα καθιστά συγκριτικά ακατάλληλα;