Project 9 – Data Engineering

**Περιγραφή**:

Σκοπός της συγκεκριμένης εργασίας είναι ο πειραματισμός μεθόδων “*data engineering*” πάνω σε πραγματικά δεδομένα. Πιο συγκεκριμένα, σας δίνεται το dataset [bankloan.csv](https://drive.google.com/file/d/1MSQC1B-Zn5ZmoNHjY3jNoRkiePAd4E6Z/view?usp=sharing) το οποίο περιέχει αληθινά δεδομένα δανειοληπτών. Στο dataset περιλαμβάνονται 53 χαρακτηριστικά, των οποίων η περιγραφή βρίσκεται στον παρακάτω σύνδεσμο: <https://www.kaggle.com/datasets/hetvigandhi03/loan-risk-analysis-dataset-real-world-data>. Για την άσκηση αυτή θα υλοποιήσετε μια εφαρμογή «Risk Analysis» (Που κάνουν οι τράπεζες, ασφαλιστικές εταιρίες, κλπ.), η οποία θα αξιολογεί τους υποψήφιους δανειολήπτες, με στόχο τη χορήγηση δανείων στους καλύτερους από αυτούς (όχι σε όλους). Συνήθως, οι τράπεζες βαθμολογούν τους υποψηφίους με ένα βαθμό (grade) από το A ως το G (A > B > C > D > F > G). και ξεκινούν να χορηγούν δάνεια με τη σειρά (Πρώτα τους Α, μετά τους Β κλπ.), επομένως είναι λογικό αρχικά να ελέγξουν τις αιτήσεις όλων των υποψηφίων.

Αν κάποια τράπεζα δέχεται πολλές αιτήσεις για επιδότηση δανείου, τότε η διαδικασία δανειοδότησης (Εικόνα 1) μπορεί να διαρκέσει πάρα πολύ χρόνο. Επομένως, είναι αρκετά επιθυμητή η υλοποίηση μιας μεθοδολογίας που να αξιολογεί σε πολύ γρήγορο χρόνο τους επιθυμητούς υποψήφιους προς δανειοδότηση.

Εικόνα που περιέχει κείμενο, διάγραμμα, στιγμιότυπο οθόνης, γραμμή

Περιγραφή που δημιουργήθηκε αυτόματα

*Figure 1: Διαδικασία λήψης Δανείου από την τράπεζα*

**Ερώτημα 1**:

Περιγράψτε τις τιμές των χαρακτηριστικών (describe). Καθώς τα χαρακτηριστικά είναι πολλά, μπορείτε να χρησιμοποιήσετε την βιβλιοθήκη [ydata-profiling](https://docs.profiling.ydata.ai/latest/) η οποία δέχεται ως είσοδο ένα *pandas dataframe* και αυτομάτως:

1. Καταλαβαίνει τον τύπο των χαρακτηριστικών (Διακριτές, Συνεχής, Ονομαστικές μεταβλητές, Χρονοσειρές, κλπ.)
2. Εντοπίζει Skewed Distributions, Missing Values, Correlated Data, Imbalanced Data, κλπ.
3. Υπολογίζει και Προβάλει την κατανομή των δεδομένων.

Μπορείτε να συμβουλευτείτε το documentation:

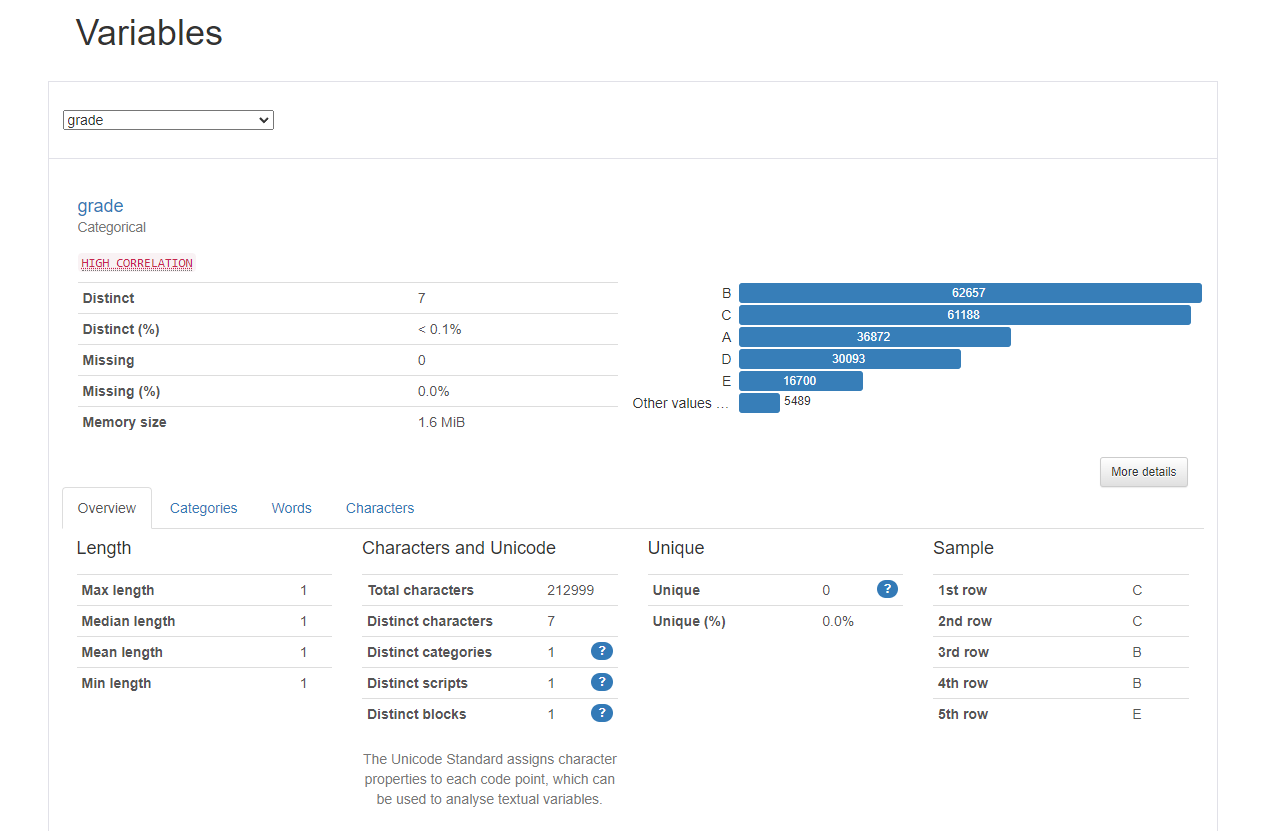
Εικόνα που περιέχει κείμενο, γραμματοσειρά, στιγμιότυπο οθόνης, γραμμή

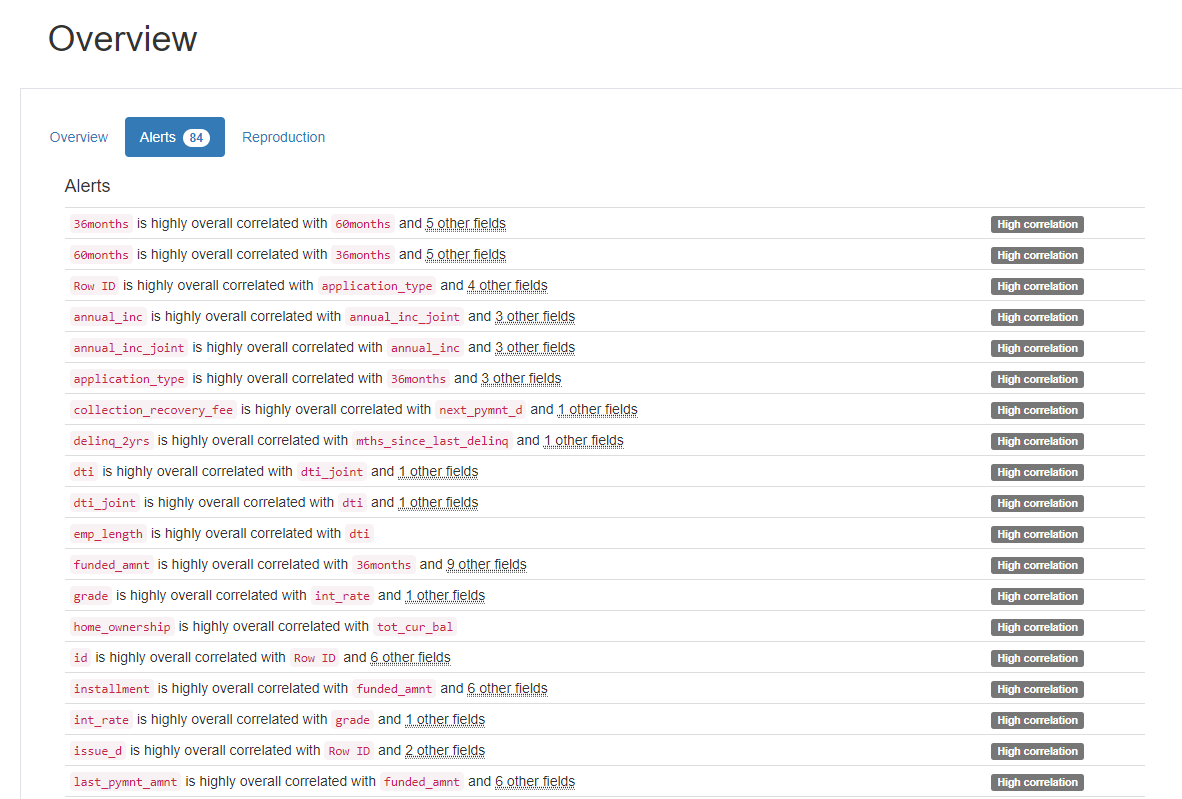
Περιγραφή που δημιουργήθηκε αυτόματα

*Figure 2: Data Profiling in Python*

Σημείωση: Υπάρχει περίπτωση να μην μπορέσει το colab να φορτώσει το ydata-profiling λόγο μεγάλου όγκου δεδομένων. Σε αυτή τη περίπτωση μπορείτε να εξάγεται το profiling ως αρχείο html (profile.to\_file("report.html")) και στη συνέχεια να το κατεβάσετε από το colab και να το ανοίξετε μέσω του αγαπημένου σας φυλλομετρητή. Αν παρόλα αυτά δεν τα καταφέρετε, μπορείτε να το κατεβάσετε από εδώ: <https://drive.google.com/file/d/149Vfesp33UPxh8-N4deVolx-3yDa4ppK/view?usp=sharing>

**Παράδειγμα**:





**Ερώτημα 2 (Data Preprocessing & Visualization)**:

Συμβουλευτείτε το dataset description και το profiling dashboard που δημιουργήθηκε από το ερώτημα (1), ώστε να απαντήσετε τα ερωτήματα:

1. Ποιο είναι το μέσο, μέγιστο, ελάχιστο ποσό *loan\_amnt* που αιτούνται οι υποψήφιοι στο σύνολο δεδομένων?
2. Ποιες μεταβλητές μπορούμε να αφαιρέσουμε για την εκπαίδευση ενός μοντέλου? **Αφαιρέστε τις μεταβλητές που δεν χρειάζονται.**
3. Εφαρμόστε όποια άλλη μέθοδο preprocessing θεωρείτε απαραίτητη.
4. Έστω η τράπεζα αποφασίζει να χορηγήσει δάνεια στους Α, Β1, Β2 και ως target θεωρήσουμε αν κάποιος υποψήφιος θα λάβει δάνειο ή όχι (Δηλαδή αν θα βρίσκεται στις κατηγορίες Α, Β1 ή Β2) τι συμπεραίνετε για τα targets? Δημιουργήστε αντίστοιχη στήλη για τα targets.
5. Υπάρχει κάποιο αιτούμενο **εύρος ποσού (πχ από 20000-25000)** (*loan\_amnt*) το οποίο να γίνει δεκτό για δανειοδότηση με πιθανότητα τουλάχιστον από αυτούς που το αιτούνται? Αν Ναι, να βρείτε και να αναφέρετε με κατάλληλο τρόπο ένα τέτοιο εύρος.

**Ερώτημα 3 (Classification)**:

Χρησιμοποιείστε κάποιον Classifier για να προβλέψετε αν κάποιος υποψήφιος είναι καλός για δανειοδότηση (σύμφωνα με το ερώτημα 2.4). Θα πρέπει:

1. Να κανονικοποιήσετε (normalization) τις εισόδους με κατάλληλο τρόπο. Είστε ελεύθεροι να επιλέξετε μέθοδο, αλλά θα πρέπει να τη δικαιολογήσετε.
2. Να χωρίσετε κατάλληλα το dataset σε train, test sets σε (70-30). Χρησιμοποιείστε τη μέθοδο Stratify για τον διαχωρισμό.
3. Εφαρμόστε κατάλληλο classifier για την πρόβλεψη της δανειοδότησης ενός υποψηφίου. Μπορείτε να χρησιμοποιήσετε όποιο μοντέλο θέλετε, αρκεί να δικαιολογήσετε την επιλογή του. Επιπλέον, αν θέλετε, μπορείτε να εφαρμόσετε cross-validation στο train set για την εύρεση κατάλληλων παραμέτρων του μοντέλου σας. Επειδή το dataset είναι μεγάλο, επιλέξτε μόνο την πιο ευαίσθητη παράμετρο (προαιρετικά). **ΠΡΟΣΟΧΗ**: **Υπάρχει κάποια μεταβλητή που προδίδει το target, καθώς η τράπεζα αποδίδει και την κατηγορία του υποψηφίου και τη τιμή αυτής της μεταβλητής. Βρείτε την μεταβλητή αυτή και αφαιρέστε την.**
4. Αναφέρετε την ακρίβεια, F1, Precision, Recall για το καλύτερο μοντέλο του validation set. Ποιες μετρικές είναι πιο σημαντικές για αυτή την εφαρμογή? Δικαιολογείστε.

**Ερώτημα 4 (Feature Importance & Selection)**:

1. Εκπαιδεύστε ταξινομητή Random Forest (εφόσον εφαρμόσετε κανονικοποίηση και διαχωρισμό σε train/test) και αναφέρετε τα 15 σημαντικότερα χαρακτηριστικά στη πρόβλεψη δανειοδότησης.
2. Τυπώστε το correlation matrix για αυτά τα χαρακτηριστικά. Υπάρχει correlation μεταξύ των χαρακτηριστικών αυτών?
3. Κρατήστε κάποια χαρακτηριστικά που να έχουν χαμηλό correlation μεταξύ τους και Επαναλάβετε το ερώτημα 1.

**Ερώτημα 5 (Dimensionality Reduction)**:

1. Φορτώστε το dataset των ασπρόμαυρων ψηφίων MNIST (της προηγούμενης άσκησης).
2. Εφαρμόστε τη μέθοδο Standard Scaling στα δεδομένα.
3. Εφαρμόστε PCA, ώστε να μειώσετε τα χαρακτηριστικά σε 300. Στη συνέχεια, εκπαιδεύστε ταξινομητή KNN με *k=5,15,51, 101*. **Προσοχή**: **Θέλουμε να καλέσουμε την** **pca.fit\_transform για το train set, αλλά μόνο τη συνάρτηση pca.transform για το test set.** Να αιτιολογήσετε τι θα γινόταν αν εφαρμόζαμε την fit transform στα x\_train και x\_test ξεχωριστά, ή σε όλα μαζί (δηλαδή αν ενώναμε τα x\_train, x\_test και εφαρμόζαμε pca) και υπολογίστε την ακρίβεια του ταξινομητή για κάθε *k.*
4. Εφαρμόστε LDA μειώνοντας τα χαρακτηριστικά σε 2 (lda.fit\_transform στο x\_train και lda.transform στο test). Στη συνέχεια: Να δημιουργήσετε 2D scatter plot με τα x\_train ύστερα από σχηματισμό LDA. Δώστε διαφορετικό χρώμα σε κάθε ψηφίο. Τι παρατηρείτε για τον διαχωρισμό τους?
5. Επαναλάβεται το ερώτημα 4, χρησιμοποιώντας τη βιβλιοθήκη umap:<https://umap-learn.readthedocs.io/en/latest/> (ενδεχομένως να χρειαστεί εγκατάσταση στο colab: !pip install umap-learn). Να συγκρίνεται το διαχωρισμό της μεθόδου U-MAP με τον αντίστοιχο της LDA.