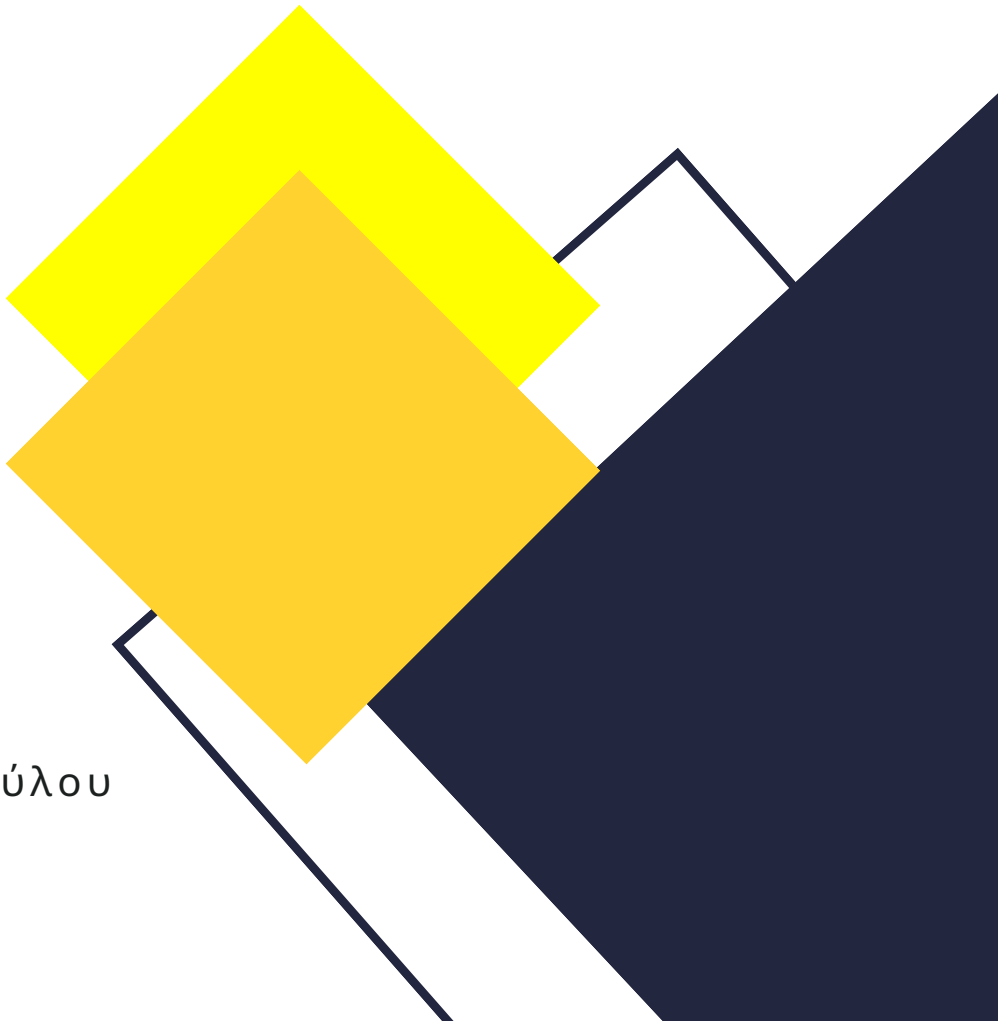
ΕΞΟΡΥΞΗ ΔΕΔΟΜΕΝΩΝ ΚΑΙ ΑΛΓΟΡΙΘΜΟΙ ΜΑΘΗΣΗΣ

2021 / 2022



Authors:

Μάριος Στεφανίδης
- 1067458

Αικατερίνη Μητροπούλου
- 1067409

Άσκηση 1

Ερώτημα Α

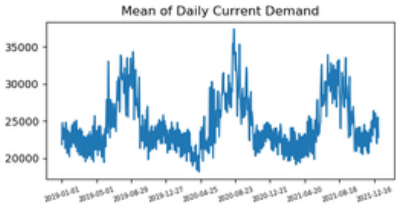
Στο συγκεκριμένο ερώτημα έχουν δημιουργηθεί δύο μεγάλα dataframes (demand και sources) από την ενοποίηση των επιμέρους csv αρχείων που μας έχουν δοθεί. Ωστόσο, κατά τη διάρκεια της παραπάνω διαδικασίας, πραγματοποιήθηκε κάποιο pre-processing στα δεδομένα και πιο συγκεκριμένα:

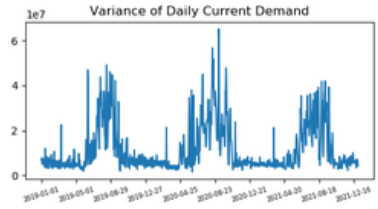
- έχει γίνει έλεγχος για ύπαρξη κενών csv
- ο τίτλος κάθε στήλης csv μετατράπηκε σε πεζά γράμματα, προκειμένου να αποφύγουμε τη δημιουργία πολλαπλών ίδιων κατηγοριών
- στα δύο τελικά dataframes έχει προστεθεί και μία επιπλέον στήλη που περιέχει κάθε φορά την ημερομηνία, στην οποία αναφέρεται κάθε γραμμή (έχει γίνει και ταξινόμηση των dataframes σύμφωνα με την ημερομηνία κατά αύξουσα σειρά)
- διαγραφή της τελευταίας σειράς κάθε csv, αφού ανεφερόταν σε ώρα της επόμενης μέρας (00:00)
- αντικατάσταση NaN values με την τιμή μηδέν
- αφαίρεση διπλότυπων σειρών σε κάθε csv

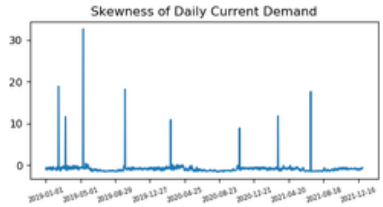
Για την ευκολότερη και γρηγορότερη εκτέλεση του κώδικα κάθε αποτέλεσμα που προέκυπτε (όπως φαίνεται και παρακάτω) έχει αποθηκευτεί σε μορφή csv και περιέχεται στο παρόν .zip αρχείο (βλ. sources και demand που αποτελούν τα ενοποιημένα csv που αναφέρθηκαν και παραπάνω).

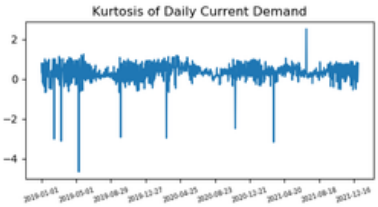
Στη συνέχεια, αποφασίσαμε να επικεντρωθούμε κυρίως στα demands, συνεπώς έχουν δημιουργηθεί συναρτήσεις που υπολογίζουν τα βασικά στατιστικά μεγέθη (μέσος όρος, διακύμανση, κύρτωση και λοξότητα) της στήλης current demand κάθε μέρας, κάθε τριμήνου και κάθε χρόνου.

Παρακάτω παρουσιάζονται οι γραφικές παραστάσεις των μεγεθών που αναφέρθηκαν παραπάνω με τα αντίστοιχα χρονικά διαστήματα.

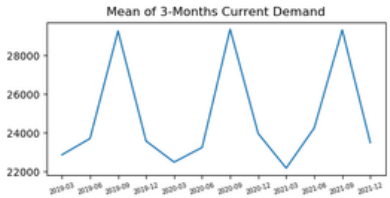


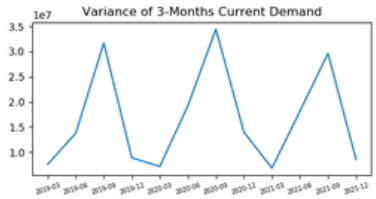


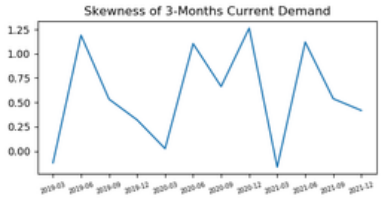


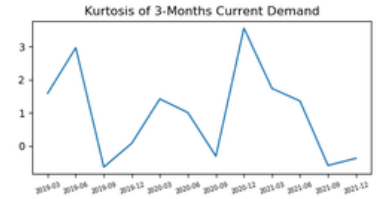


Στατιστικά μεγέθη των demands για κάθε μέρα (statistics.csv)

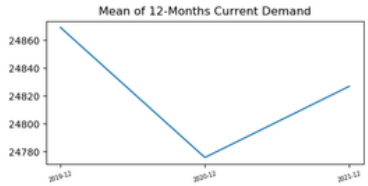


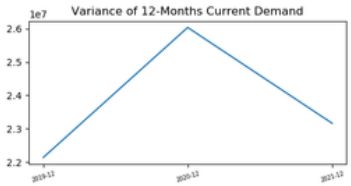


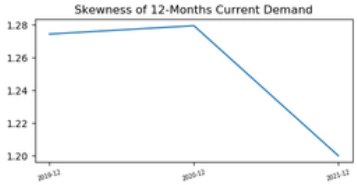


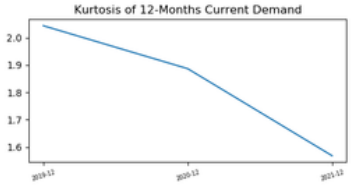


Στατιστικά μεγέθη των demands για κάθε τρίμηνο (statistics3.csv)





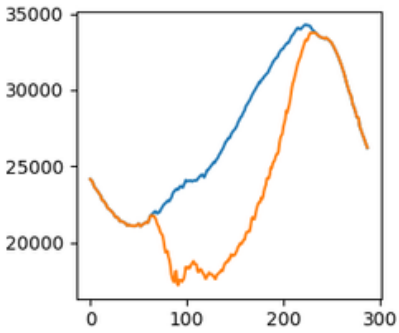


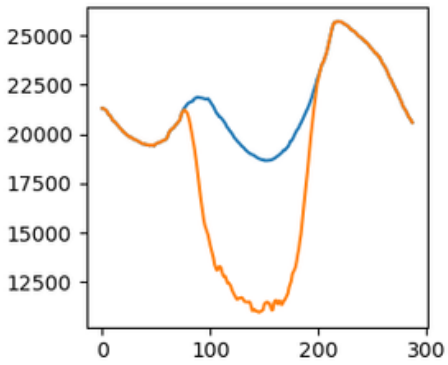


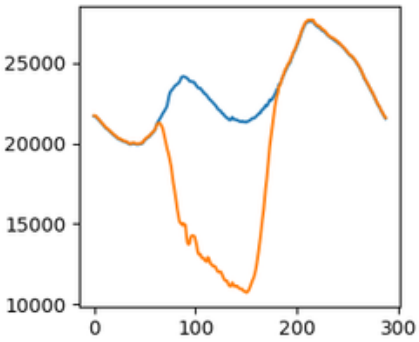
Στατιστικά μεγέθη των demands για κάθε χρόνο (statistics12.csv)

Ακόμη, σχεδιάσαμε και το duck curve, το οποίο ουσιαστικά είναι ένα γράφημα παραγωγής ενέργειας κατά τη διάρκεια μιας μέρας που δείχνει τη χρονική ανισορροπία μεταξύ της ζήτησης αιχμής και της παραγωγής ανανεώσιμης ενέργειας.

Παρακάτω, παρουσιάζονται ενδεικτικά κάποια γραφήματα για τυχαίες μέρες.







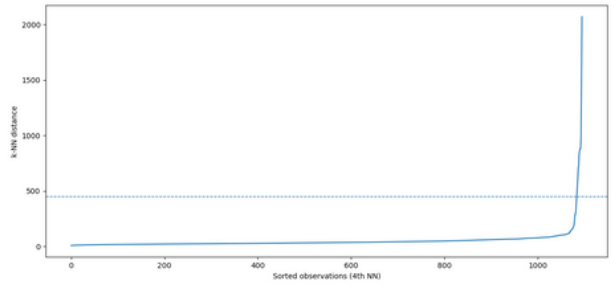
Σύμφωνα με τα duck curves παρατηρούμε πως η χρήση της ηλιακής ενέργειας είναι καίριας σημασίας ιδιαίτερα τις μεσημεριανές ώρες, αφού φαίνεται πως δεν απαταιτείται ιδιαίτερα η χρήση των υπόλοιπων demands. Αυτό σημαίνει πως εκείνες τις χρονικές περιόδους η ηλιακή ενέργεια καλύπτει ένα πολύ μεγάλο ποσοστό τω αναγκών των πολιτών της California, κάτι βέβαια που αναμέναμε. Ακόμα, σύμφωνα με τις παραπάνω γραφικές παραστάσεις των στατιστικών μεγεθών παρατηρούμε πως από τον έκτο εώς περίπου των ένατο μήνα κάθε χρόνου τα demands των πολιτών αυξάνονται κατά πολύ, κάτι βέβαια που είναι απολύτως λογικό, αν σκεφτεί κάποιος το κλίμα που χαρακτηρίζει τη συγκεκριμένη πολιτεία.

Ερώτημα Β

Στο συγκεκριμένο ερώτημα αποφασίσαμε να ομαδοποιήσουμε τις τιμές της ζήτησης και των διαθέσιμων πόρων κάθε μέρας, προκειμένου να εντοπίσουμε outliers, τιμές δηλαδή κατά τις οποίες η ζήτηση ή η παραγωγή δεν είχαν τις αναμενόμενες τιμές. Πιο συγκεκριμένα, ετοιμάστηκε το παρακάτω dataset, που δόθηκε ως είσοδος:

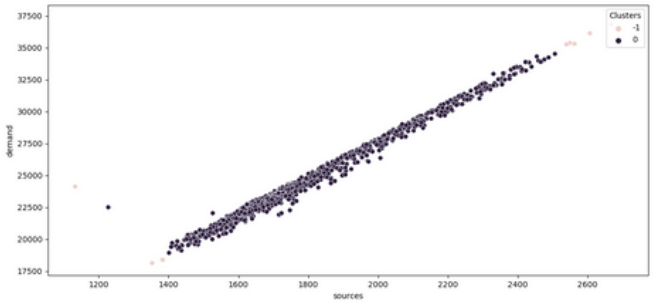
- μέσος όρος της στήλης current demand κάθε μέρας του ενοποιημένου αρχείου demand.csv που έχει υπολογιστεί από το προηγούμενο ερώτημα
- έχουν ακόμα υπολογιστεί οι μέσοι όροι κάθε πηγής για κάθε μέρα και στη συνέχεια υπολογίστηκαν οι μέσοι όροι των πηγών που διατίθενται κάθε μέρα, οι οποίοι δόθηκαν τελικά και ως είσοδος

Αποφασίσαμε να εφαρμόσουμε την μέθοδο clustering, **DBSCAN** μιας και δεν απαιτεί από κάποιον να καθορίσει τον αριθμό των clusters στα δεδομένα εκ των προτέρων σε αντίθεση με το k-means. Επίσης, το DBSCAN μπορεί να βρει clusters αυθαίρετου σχήματος ενώ ακόμα μπορεί να βρει ένα cluster, το οποίο περιβάλλεται εντελώς από ένα διαφορετικό cluster. Προκειμένου να καθορίσουμε το ε, εφαρμόσαμε το elbow method, το οποίο ουσιαστικά είναι μια ευρετική μέθοδος που χρησιμοποιείται για τον προσδιορισμό συστάδων σε ένα σύνολο δεδομένων.



Σύμφωνα με τη γραφική παράσταση επιλέχθηκε για ε = 400

Σύμφωνα με τη μέθοδο DBSCAN λοιπόν προέκυψε το παρακάτω clustering:

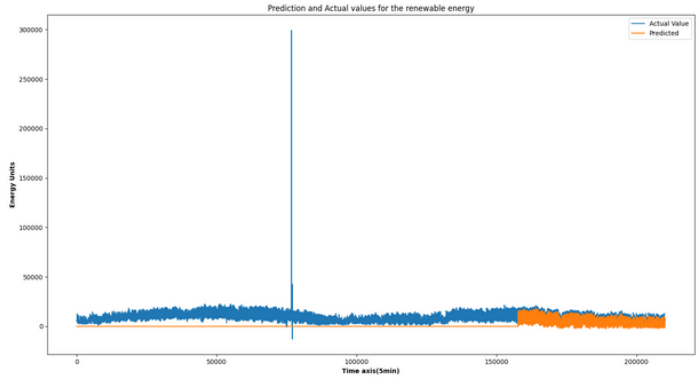


-1 -> outliers
0 -> συστάδα

Παρατηρούμε από την γραφική παράσταση πως προκύπτει μία συστάδα ενώ υπάρχουν 9 outliers που αναφέρονται στις ημερομηνίες: 2019-01-02, 2020-04-11, 2020-04-12, 2020-04-19, 2020-08-14, 2020-08-15, 2020-08-17, 2020-08-18, 2020-08-19 και 2020-09-06

Ερώτημα Γ

Στο συγκεκριμένο ερώτημα προσπαθήσαμε να εκπαιδεύσουμε έναν παλινδρομητή βασισμένο σε LSTM νευρωνικά δίκτυα, ο οποίος προσπαθεί να προβλέψει για κάθε στιγμή της ημέρας πόση ενέργεια απαιτείται να παραχθεί από τη χρήση μη-ανανεώσιμων πηγών ενέργειας.
Αρχικά, αφαιρέσαμε τους outliers που εντοπίσαμε στο παραπάνω ερώτημα, από το dataset ενώ χωρίσαμε το τελευταίο σε train (75%) και test set (25%). Έπειτα, εφαρμόσαμε min-max transform στα δεδομένα σε ένα εύρος στόχου μεταξύ [0, 1] και προχωρήσαμε με την κατασκευή του LSTM Network.
Αντιμετωπίσαμε ένα μικρό πρόβλημα αναφορικά με τις παραμέτρους που έπρεπε να λάβουμε υπόψιν μας και να συμπεριλάβουμε τελικά στο μοντέλο, καθώς επίσης και με τις εποχές (epochs). Ωστόσο, καταλήξαμε στην παρακάτω πρόβλεψη:



Παρατηρούμε πως η πρόβλεψη του μοντέλου είναι αρκετά καλή, ωστόσο υπάρχει μεγάλη πιθανότητα για overtraining του μοντέλου (δηλαδή μπορεί να προβλέψει training set με πολύ υψηλή ακρίβεια, αλλά ενδεχόμενως να μην είναι τόσο ακριβές σε νέα δεδομένα).

Άσκηση 2

Σκοπός του συγκεκριμένου ερωτήματος είναι να προσπαθήσουμε να μαντέψουμε τη βαθμολογία προϊόντων μέσω διαφόρων κριτικών χρηστών που έχουν υποβάλει για αυτά (amazon.csv). Θα χρησιμοποιήσουμε ένα νευρωνικό δίκτυο (και πιο συγκεκριμένα έναν RandomForest), στο οποίο θα δοθεί ως είσοδος κείμενα υπό την μορφή διανυσμάτων (τεχνική Word Embeddings).

Word Embeddings

Για αρχή, θα χρησιμοποιήσουμε την ενσωματωμένη συνάρτηση που παρέχει το gensim προκειμένου να χειριστούμε τον καθαρισμό και το tokenization των δεδομένων (κατάργηση σημείων στίξης, αφαίρεση stop words κ.ο.κ.). Στη συνέχεια, θα χωρίσουμε το dataset σε train (80%) και test set (20%) για το νευρωνικό δίκτυο που αναλύεται παρακάτω.

Έπειτα από την κατάλληλη επεξεργασία των κριτικών κάθε χρήστη (διαχωρισμός κάθε κριτικής σε προτάσεις και κάθε πρότασης σε λέξεις), εφαρμόζουμε το Word2Vec μοντέλο.

RandomForest

Κάθε λέξη αποτελείται από έναν πίνακα, συνεπώς κάθε κριτική είναι μια σειρά από πίνακες. Αυτό σημαίνει πως έχει δημιουργηθεί ένας μεγάλος πίνακας που αποτελείται από υπό-πίνακες. Το πρόβλημα που προκύπτει είναι πως κάθε πρόταση θα έχει προφανώς διαφορετικό αριθμό πινάκων από διανύσματα, γεγονός που μπορεί να προκαλέσει σφάλμα κατά την εκπαίδευση του μοντέλου. Για αυτό το λόγο υπολογίσαμε το μέσο όρο όλων των λέξεων από τις οποίες αποτελείται κάθε κριτική του dataset που διαθέτουμε. Αφού δημιουργηθούν λοιπόν οι παραπάνω πίνακες διανυσμάτων, δίνονται ως είσοδο στο μοντέλο RandomForest, τα αποτελέσματα του οποίου φαίνονται παρακάτω:

	precision	recall	f1-score	support
1	0.55	0.27	0.36	1025
2	0.44	0.01	0.03	556
3	0.44	0.04	0.07	761
4	0.34	0.06	0.10	1672
5	0.64	0.98	0.77	5986
accuracy			0.63	10000
macro avg	0.48	0.27	0.27	10000
weighted avg	0.56	0.63	0.52	10000

Recall: 0.6261

Precision: 0.6261

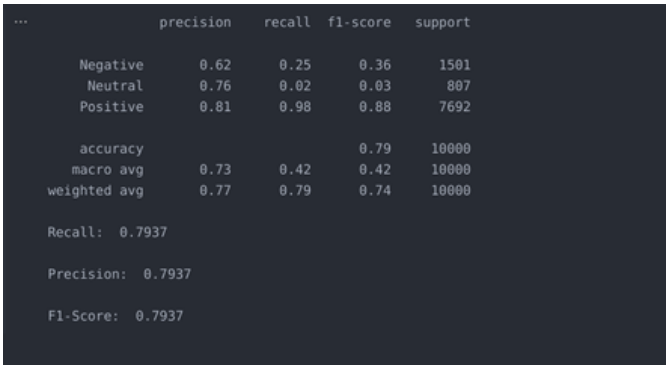
F1-Score: 0.6261

Σύμφωνα με τις μετρικές παρατηρούμε πως η απόδοση του μοντέλου είναι μέτρια μιας και επιτυγχάνει ένα accuracy 62%

Προκειμένου να πετύχουμε μεγαλύτερο accuracy, αποφασίσαμε να χωρίσουμε τις βαθμολογίες των κρητικών που προκύπτουν από τους χρήστες στις εξής κατηγορίες:

- κριτικές με score < 3 χαρακτηρίζονται ως Negative
- κριτικές με score = 3 χαρακτηρίζονται ως Neutral
- κριτικές με score > 3 χαρακτηρίζονται ως Positive

Σύμφωνα με την παραπάνω κατηγοριοποίηση το μοντέλο έχει τα παρακάτω αποτελέσματα:



	precision	recall	f1-score	support
Negative	0.62	0.25	0.36	1501
Neutral	0.76	0.02	0.03	807
Positive	0.81	0.98	0.88	7692
accuracy			0.79	10000
macro avg	0.73	0.42	0.42	10000
weighted avg	0.77	0.79	0.74	10000

Recall: 0.7937

Precision: 0.7937

F1-Score: 0.7937

Παρατηρούμε πως η απόδοση του μοντέλου είναι αρκετά ικανοποιητική (accuracy -> 80%)

Σημειώση: Έχουν επισυναπτεί τα αρχεία πηγαίου κώδικα .ipynb και .py για κάθε άσκηση.