ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ Η/Υ & ΠΛΗΡΟΦΟΡΙΚΗΣ

Εξόρυξη Δεδομένων και Αλγόριθμοι Μάθησης

Εαρινό εξάμηνο 2021-22

Αναφορά Υλοποιητικού Project

Στοιχεία ομάδας:

Παπαγεωργίου Νικηφόρος – Γεώργιος	1059633	st1059633@ceid.upatras.gr
Παπανικολάου Αικατερίνη	1064041	st1064041@ceid.upatras.gr

ΙΟΥΝΙΟΣ 2022 | ΠΑΤΡΑ

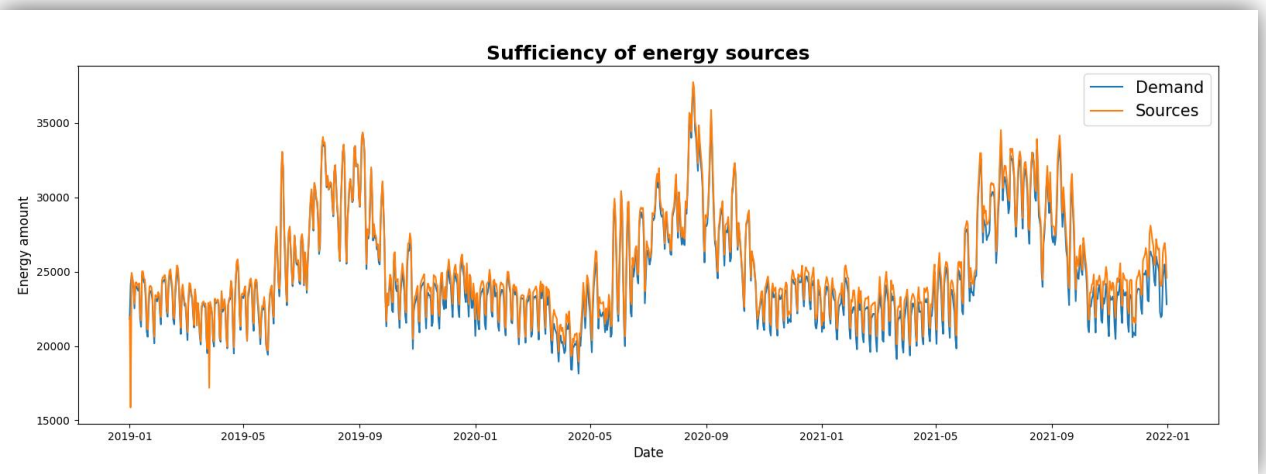
Πληροφορίες υλοποίησης

Για την υλοποίηση των ερωτημάτων της εργασίας χρησιμοποιήθηκε η γλώσσα προγραμματισμού **Python** και, πιο συγκεκριμένα, η έκδοση **3.6.8** αυτής. Χαρακτηριστικές βιβλιοθήκες που χρησιμοποιήσαμε είναι οι pandas, numpy, matplotlib, sklearn, nltk, κ.α. Το σύνολο των ερωτημάτων της εργασίας υλοποιήθηκαν στο περιβάλλον του **Jupyter Notebook** και τα αρχεία κώδικα που παραδίδουμε είναι της μορφής ".ipynb". Ακολουθεί μία σύντομη περιγραφή της υλοποίησης κάθε ζητούμενου. Αναλυτικότερη περιγραφή παρέχεται μέσω σχολίων στα αντίστοιχα αρχεία κώδικα.

Ερώτημα 1

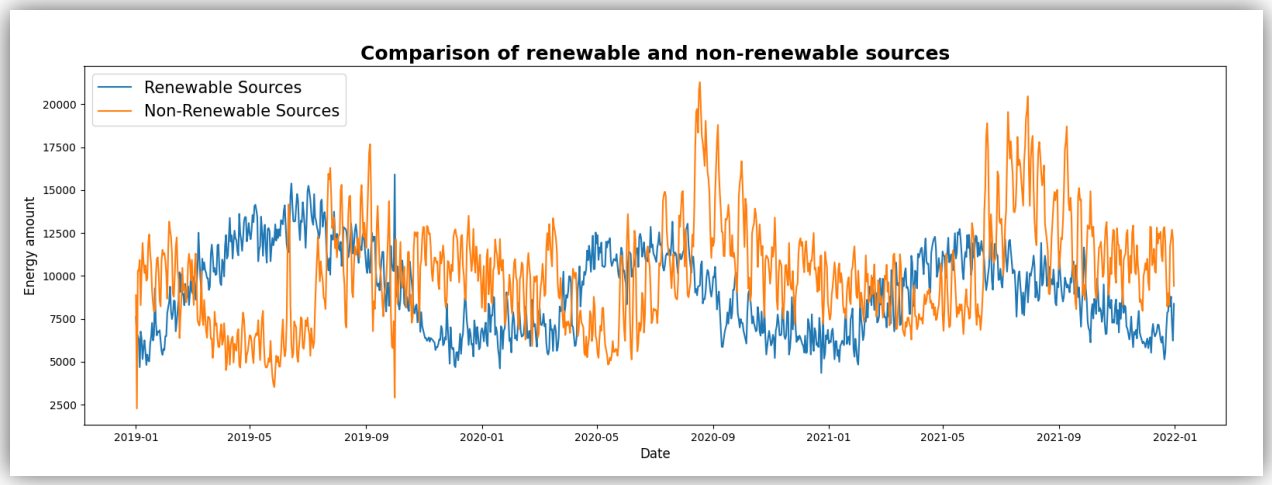
Α. Αρχικά, δημιουργούμε δύο λίστες, μία για τα αρχεία του φακέλου sources και μία για του φακέλου demand, όπου περιέχουν τα αντίστοιχα sources και demand dataframes. Στη συνέχεια, ενοποιούμε κάθε ζεύγος sources-demand dataframes, που αντιστοιχούν στην ίδια ημέρα, και παράγουμε μία τρίτη λίστα, όπου περιέχει τα ενοποιημένα dataframes. Από αυτή τη λίστα, δημιουργούμε αρχικά ένα νέο dataframe "df_days", παίρνοντας το μέσο όρο όλων των στηλών για κάθε ημέρα. Απ' αυτό το dataframe παράγουμε τα παρακάτω γραφήματα:

1. Κάνουμε plot τη στήλη της ζήτησης μαζί με τη στήλη της συνολικής ποσότητας ενέργειας από τις πηγές, και διαπιστώνουμε πως οι ενεργειακές ανάγκες της Πολιτείας της Καλιφόρνια καθ' όλο αυτό το διάστημα των δύο χρόνων καλύπτονταν πλήρως από την ενέργεια που προσέφεραν οι διαθέσιμες πηγές. Συνεπώς, υπάρχει επάρκεια πόρων, καθώς η γραφική παράσταση των Sources είναι πάντα πιο πάνω από την αντίστοιχη γραφική παράσταση του Demand.



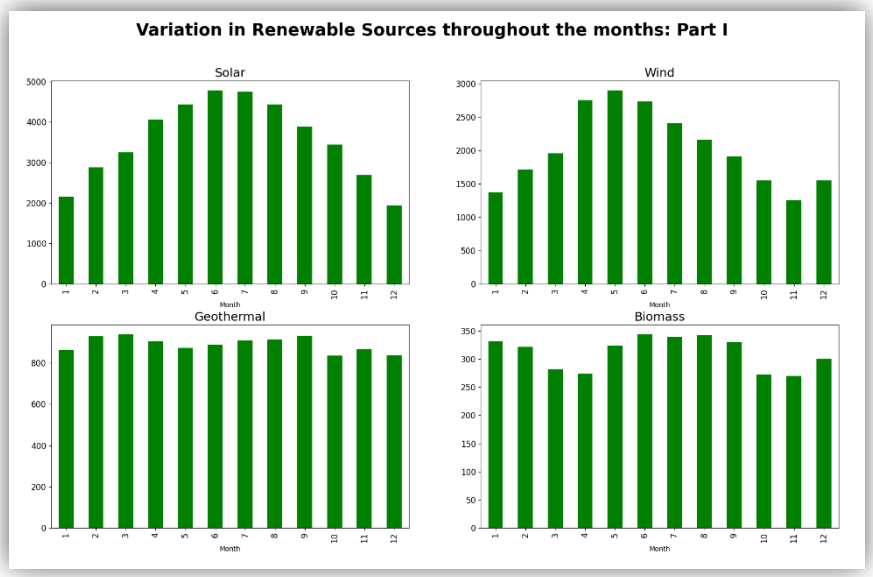
Εικόνα 1.Α.1: Γραφική παράσταση που δείχνει την κάλυψη των ενεργειακών απαιτήσεων από τις διαθέσιμες πηγές.

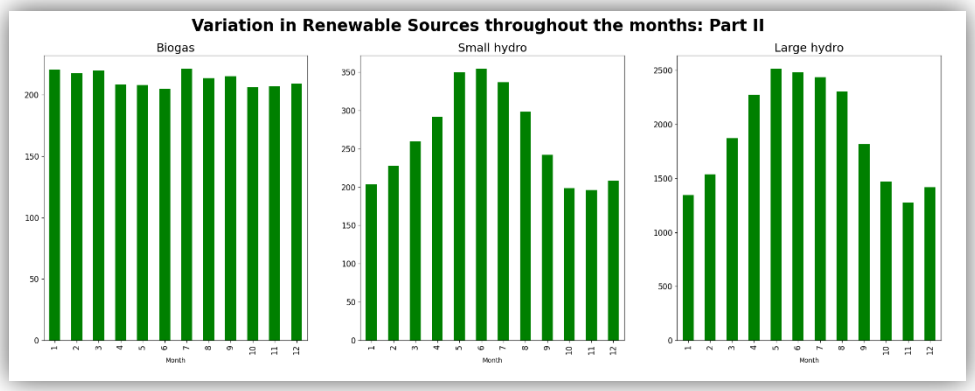
2. Κάνουμε plot το σύνολο της παρεχόμενης ενέργειας από τις ανανεώσιμες πηγές και το σύνολο από τις μη ανανεώσιμες πηγές ενέργειας. Από τις δύο γραφικές παραστάσεις παρατηρούμε πως υπάρχει μία εναλλαγή στην υπεροχή ανανεώσιμων και μη ανανεώσιμων πηγών, πράγμα το οποίο επιβεβαιώνει το γεγονός ότι επειδή οι ανανεώσιμες πηγές δεν είναι πάντα διαθέσιμες, οι επιπλέον ανάγκες καλύπτονται από τις μη ανανεώσιμες και αντίστροφα.



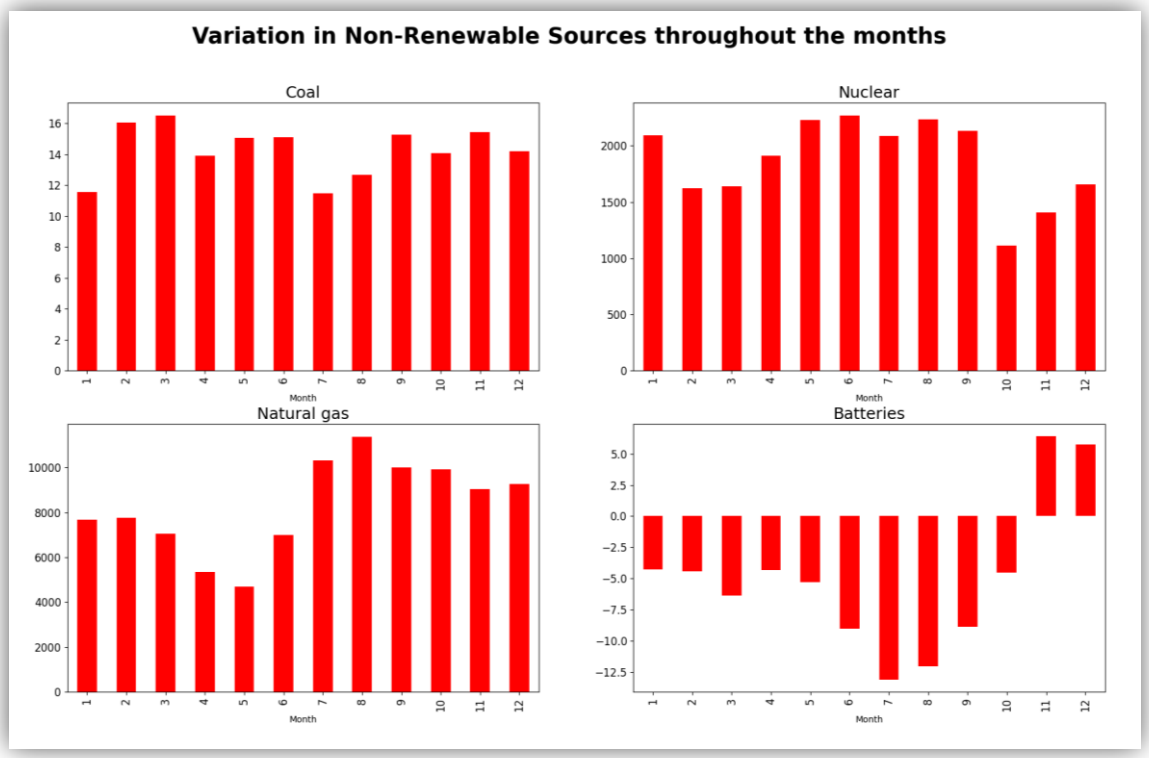
Εικόνα 1.Α.2: Γραφική παράσταση που δείχνει την παροχή ενέργειας από ανανεώσιμες και μη ανανεώσιμες πηγές.

3. Στην παρακάτω σειρά γραφημάτων, δείχνουμε τη μέση διακύμανση κάθε ανανεώσιμης και μη ανανεώσιμης πηγής κατά τη διάρκεια όλων των μηνών ενός χρόνου. Παρατηρούμε, από τις ανανεώσιμες, ότι η ηλιακή (Solar), η αιολική (Wind) και η υδροηλεκτρική (Large hydro) ενέργεια είναι αυτές που παράγουν τη μεγαλύτερη ποσότητα ενέργειας, ενώ παρατηρείται και μία γενική αύξηση των ανανεώσιμων πηγών κατά την περίοδο Άνοιξη-Καλοκαίρι. Από την άλλη, στις μη ανανεώσιμες πηγές, κυρίαρχο είναι το φυσικό αέριο (Natural gas).



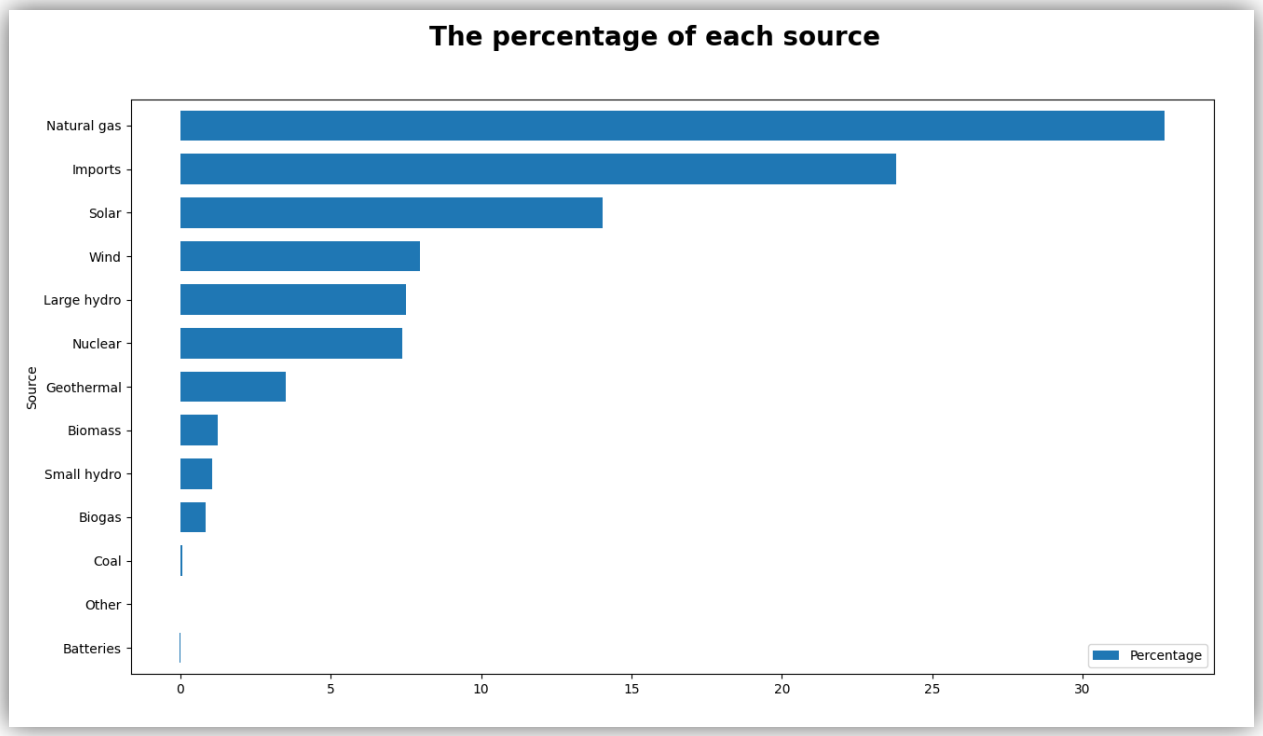


Εικόνα 1.Α.3: Bar charts διακύμανσης ανανεώσιμων πηγών.



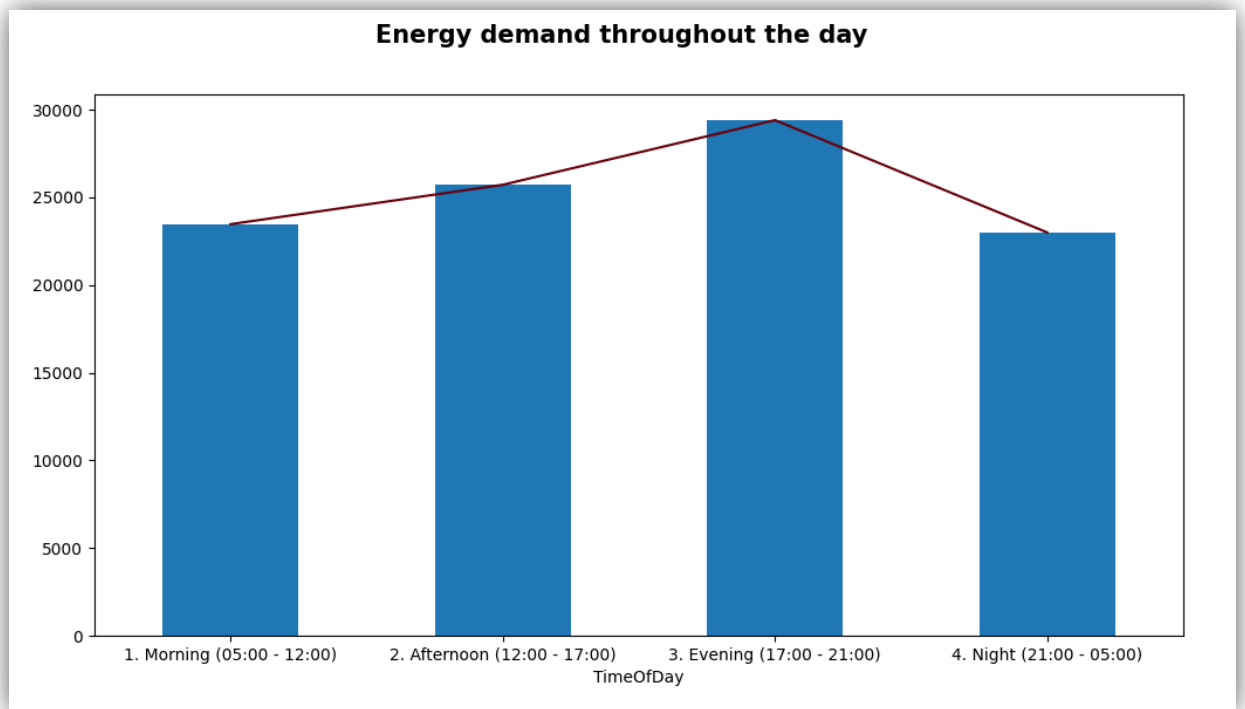
Εικόνα 1.Α.4: Bar charts διακύμανσης μη ανανεώσιμων πηγών.

4. Στο παρακάτω γράφημα, δείχνουμε σε τι ποσοστό συνεισφέρει κάθε πηγή στη συνολική ενέργεια που παράγεται.



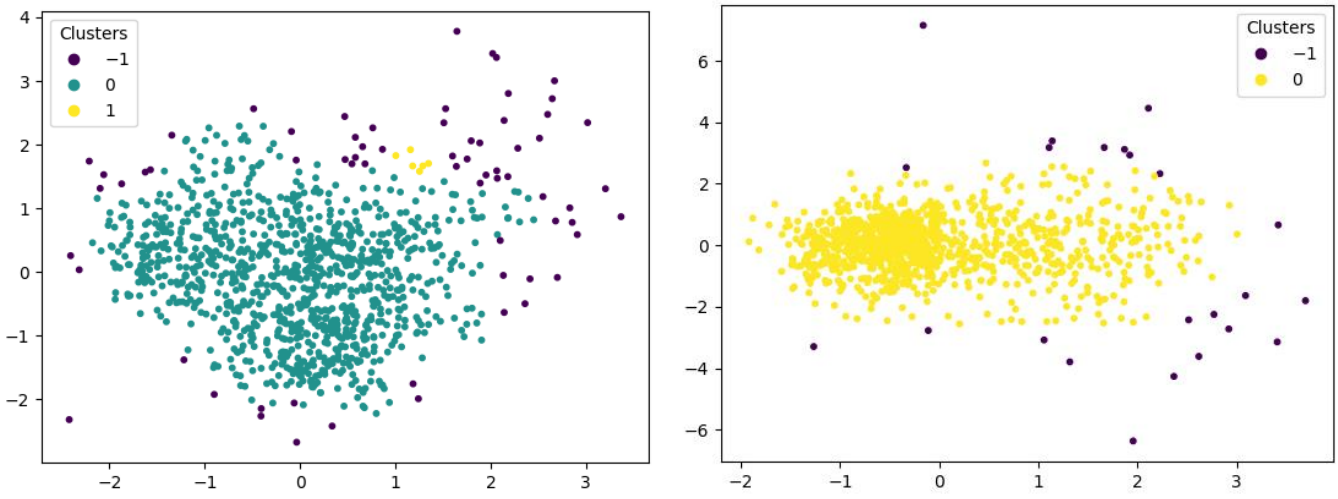
Εικόνα 1.Α.5: Ποσοστό συνεισφοράς κάθε πηγής στο σύνολο.

5. Τέλος δημιουργούμε ακόμα ένα dataframe "df_hours", το οποίο περιέχει ενεργειακά δεδομένα για κάθε 5 λεπτά της ημέρας. Αφού ομαδοποιήσουμε τις ώρες σε 4 χρονικές ζώνες, παράγουμε το ακόλουθο διάγραμμα για να δείξουμε πώς διαμορφώνεται η ποσότητα της ζήτησης καθ' όλη τη διάρκεια μίας ημέρας. Παρατηρούμε πως μεγαλύτερη ζήτηση παρατηρείται στη χρονική ζώνη 17:00-21:00.

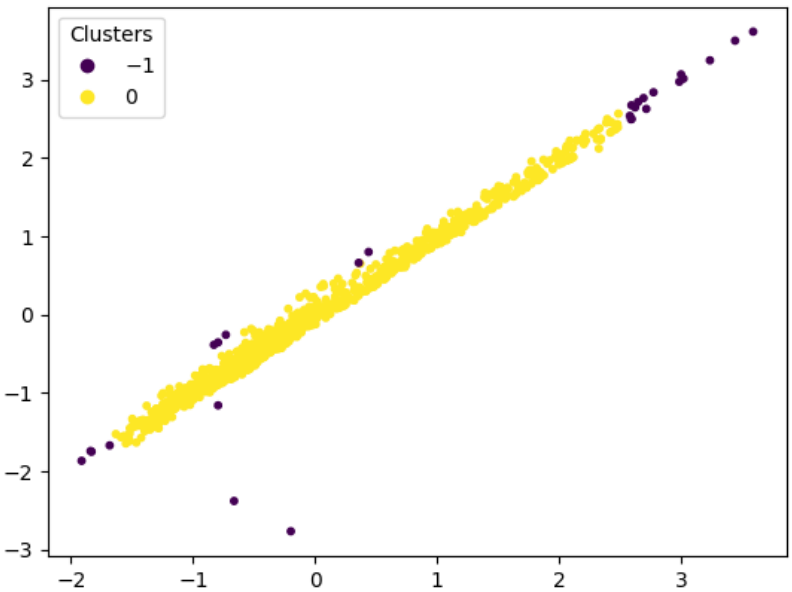


Εικόνα 1.Α.6: Πορεία της ζήτησης σε ενέργεια κατά τη διάρκεια μίας ημέρας.

B. Σ' αυτό το ερώτημα, επιλέγουμε ως αλγόριθμο ομαδοποίησης τον DBSCAN, καθώς είναι κατάλληλος στο να βρίσκει outliers σε αντίθεση με τον k-means. Αρχικά, δοκιμάζουμε να ομαδοποιήσουμε τις ημέρες ξεχωριστά με βάση την παραγωγή και ξεχωριστά με βάση τη ζήτηση, ενώ στο τέλος δοκιμάζουμε να τις ομαδοποιήσουμε με βάση συνδυασμό παραγωγής και ζήτησης. Για να βρούμε κάθε φορά την κατάλληλη παράμετρο eps για τον DBSCAN, υπολογίζουμε τη μέση απόσταση κάθε σημείου από τα 3 κοντινότερα γειτονικά του σημεία, και έπειτα κάνοντας plot τις αποστάσεις επιλέγουμε ως eps το σημείο στο γράφημα όπου υπάρχει το "elbow". Παρατηρούμε πως τα σημεία είναι αρκετά πυκνά (dense) μεταξύ τους και γι' αυτό δεν προκύπτουν πολλά clusters.



Εικόνα 1.Β.1: (α) Ομαδοποίηση ημερών βάσει παραγωγής και (β) ομαδοποίηση ημερών βάσει ζήτησης.



Εικόνα 1.Β.2: Ομαδοποίηση ημερών βάσει παραγωγής και ζήτησης.

Ερώτημα 2

Αρχικά, φορτώνουμε το αρχείο "amazon.csv" σε dataframe και προεπεξεργαζόμαστε τη στήλη Text με τις κριτικές των χρηστών, αφαιρώντας ειδικούς χαρακτήρες, σημεία στίξης, stop-words και κάνοντας στελεχοποίηση (stemming) των λέξεων που περιέχονται στις κριτικές. Έπειτα, «σπάμε» κάθε κριτική σε μία λίστα από λέξεις, την οποία δίνουμε σαν είσοδο στο Word2Vec μοντέλο που δημιουργούμε για να μας επιστρέψει το word embedding διάνυσμα κάθε κριτικής.

Στη συνέχεια, πραγματοποιούμε multi-label classification χρησιμοποιώντας έναν κατηγοριοποιητή RandomForest για να μαντέψουμε τη βαθμολογία που αντιστοιχεί σε κάθε κριτική, από το 1 έως και το 5, και εκτυπώνουμε το classification report με τις μετρικές precision, recall και f1-score για κάθε label. Παρατηρούμε πως το βέλτιστο accuracy score που πετυχαίνει ο κατηγοριοποιητής είναι 62-63% κι αυτό γιατί το dataset που μας δίνεται είναι αρκετά ανομοιόμορφο και δεν περιέχει το ίδιο πλήθος κριτικών για κάθε label, καθώς η πλειοψηφία των εγγραφών έχουν label 5. Τέλος, προκειμένου να βρούμε εκείνες τις παραμέτρους με τις οποίες ο RandomForest θα παράγει το καλύτερο δυνατό αποτέλεσμα, χρησιμοποιούμε το μοντέλο GridSearchCV το οποίο τρέχει τον RandomForest για διαφορετικούς συνδυασμούς παραμέτρων που ορίζουμε εμείς και βρίσκει τον βέλτιστο συνδυασμό βάσει accuracy score, τον οποίο και χρησιμοποιούμε εν τέλει.

	precision	recall	f1-score	support
1	0.66	0.30	0.41	1004
2	1.00	0.02	0.03	584
3	0.65	0.02	0.04	780
4	0.48	0.04	0.08	1677
5	0.63	0.99	0.77	5955
accuracy			0.63	10000
macro avg	0.69	0.27	0.27	10000
weighted avg	0.63	0.63	0.52	10000

Εικόνα 2.2: Classification report με τις μετρικές precision, recall και f1-score για κάθε label.

Σε μια προσπάθεια να βελτιώσουμε το accuracy score που περιγράψαμε παραπάνω, πραγματοποιούμε ξανά multi-label classification, αλλά αυτή τη φορά με 3 labels αντί των 5, τα οποία δείχνουν τη θετικότητα κάθε κριτικής: «**0 – Αρνητική**» (labels 1 και 2), «**1 – Ουδέτερη**» (label 3), «**2 – Θετική**» (labels 4 και 5). Κατ' αυτό τον τρόπο προσπαθούμε να αυξήσουμε το support κάθε label. Παρατηρούμε πως η ποιότητα της κατηγοριοποίησης βελτιώνεται αισθητά, με accuracy score 79-80%.

	precision	recall	f1-score	support
Negative	0.78	0.25	0.38	1588
Neutral	0.93	0.02	0.03	780
Positive	0.80	0.99	0.89	7632
accuracy			0.80	10000
macro avg	0.84	0.42	0.43	10000
weighted avg	0.81	0.80	0.74	10000

Εικόνα 2.2: Classification report με τις μετρικές precision, recall και f1-score για κάθε είδος κριτικής.