Εργασία στο μάθημα: «Εξόρυξη Δεδομένων και Αλγόριθμοι Μάθησης»

Ανδρονίκη Πορτοκάλογλου, ΑΜ:1067539
Βασίλης Μηλιώνης, ΑΜ: 1067415

Περιεχόμενα

Περιβάλλον Υλοποίησης:

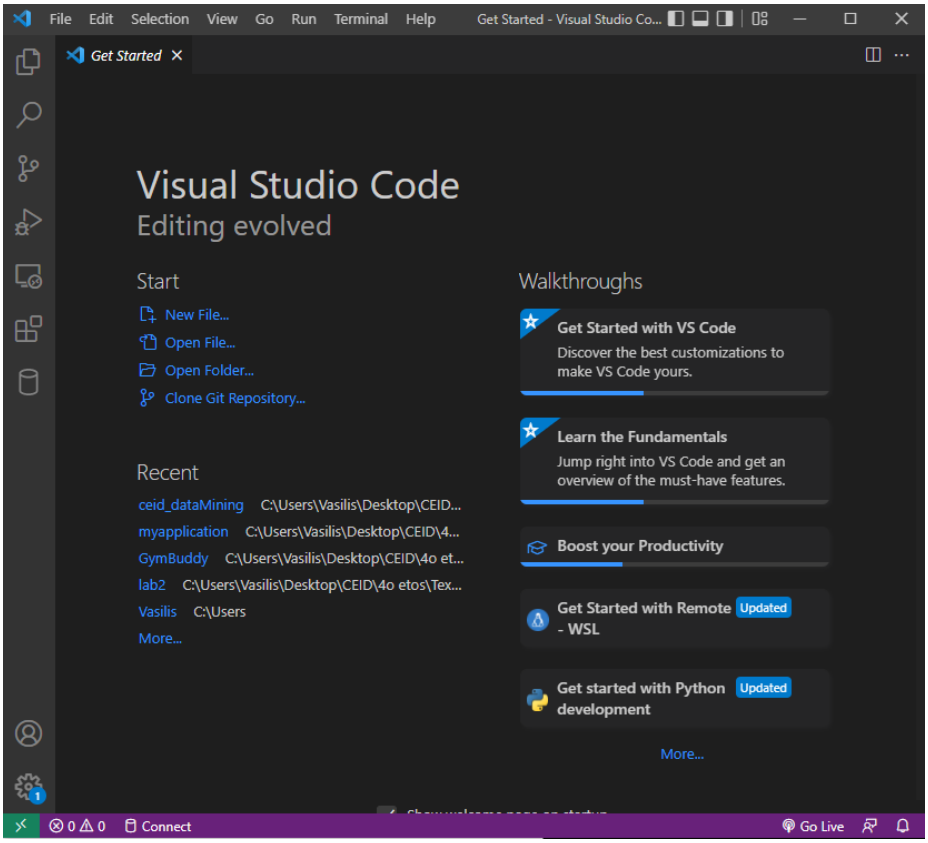
Για την ανάπτυξη των ασκήσεων χρησιμοποιήσαμε το εργαλείο jupyter. Σε αυτό μπορείτε να έχετε πρόσβαση εύκολα και μέσω του visual studio code (vscode). Από το link μπορείτε να κατεβάσετε τον editor και αφού ολοκληρωθεί η εγκατάστασή του, θα πρέπει να προσθέσετε το jupyter extension. Για αυτό το σκοπό, στην οθόνη που φαίνεται στην Εικόνα 1 πατήστε ctrl+shift+x, για να οδηγηθείτε στην καρτέλα με τα extension. Εκεί πρέπει να αναζητήσετε για τη λέξη jupyter, οπότε θα σας εμφανίσει όλα τα extension με αυτό το όνομα, όπως φαίνεται στην

Εικόνα 2. Θα πρέπει να κατεβάσετε τα τρία πρώτα extension που φαίνονται στην εικόνα αυτή. Στο δικό μου σύστημα, κατεβάζοντας το πρώτο από αυτά κάνει install και τα άλλα δύο αυτόματα. Μόλις αυτά κατέβουν έχετε ολοκληρώσει τα βήματα ρύθμισης του περιβάλλοντος jupyter.

Τώρα είστε σε θέση να ανοίξετε το φάκελο στο vscode, όπου υπάρχουν τα notebook, στον φάκελο *src/*, καθώς και τα δεδομένα που μας δόθηκαν για την υλοποίηση, στον φάκελο *resources/*. Για να το κάνετε αυτό, πατήστε ctrl+k και ctrl+o (διαδοχικά), βρείτε τον φάκελο των παραδοτέων μας και πατήστε Select Folder.

Αφού ολοκληρώσετε όλα τα βήματα της εγκατάστασης θα πρέπει να κατεβάσετε μέσω του package manager pip (θεωρούμε ότι υπάρχει στο σύστημά σας) τις βιβλιοθήκες που χρησιμοποιούμε στο project. Πιο συγκεκριμένα μπορείτε να χρησιμοποιήσετε το integrated terminal που υπάρχει στο vscode, πατώντας το συνδυασμό ctrl+shift+` για να εκτελέσετε τις εντολές:

1. *pip install matplotlib*
2. *pip install pandas*
3. *pip install seaborn*
4. *pip install numpy*
5. *pip install scikit-learn*
6. *pip install torch*
7. *pip install genism*

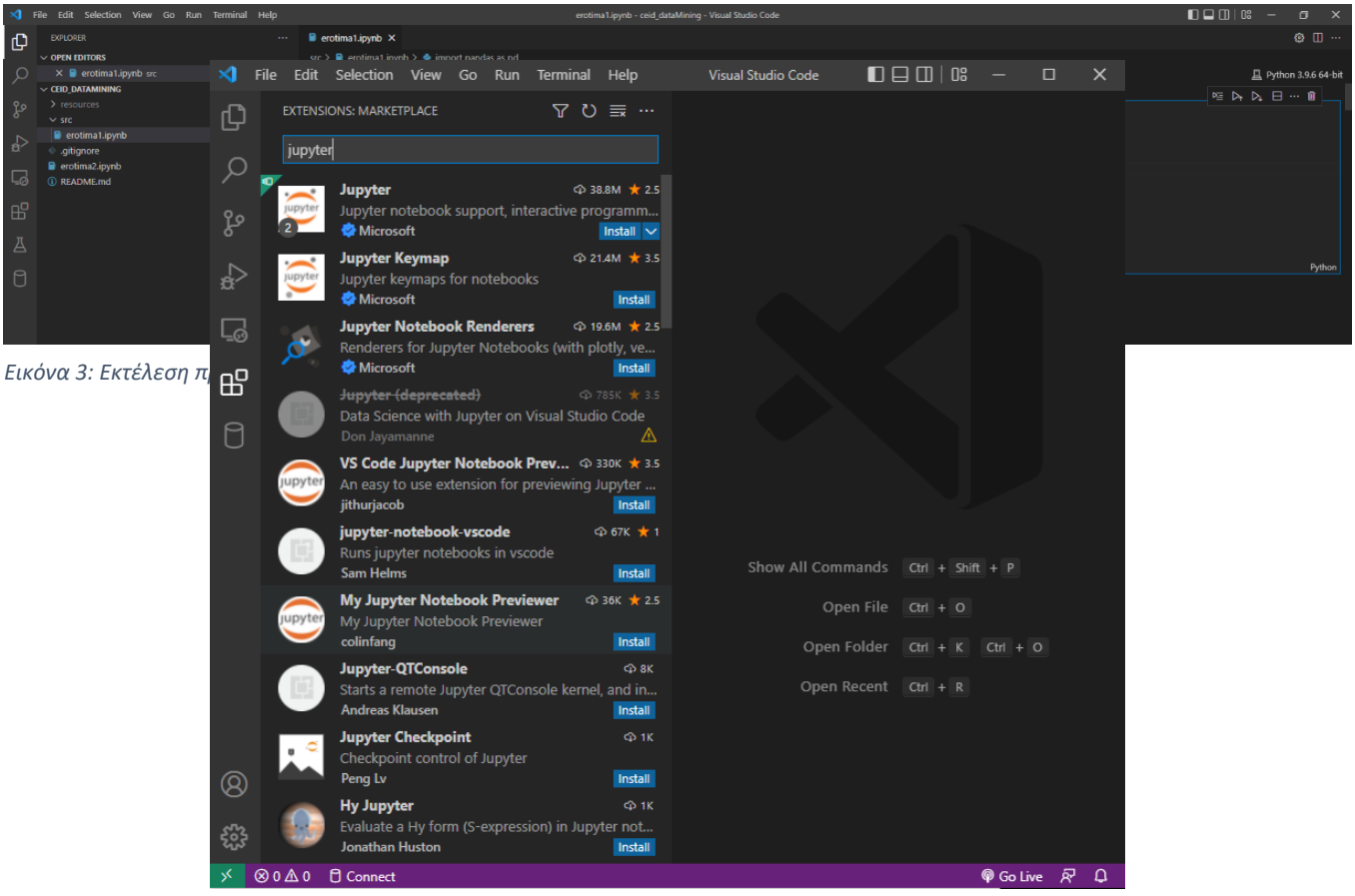


8. *pip install nlkt*

Είστε πλέον σε θέση να τρέξετε τους κώδικές μας. Ανοίξτε ένα από τα αρχεία *erotima1.ipynb* ή *erotima2.ipynb* και πατήστε Run all (κουμπί με την κόκκινη υπογράμμιση στην Εικόνα 3). Μπορείτε να τρέξετε κάθε cell κώδικα ξεχωριστά, πατώντας το πλήκτρο που έχει σημειωθεί με κίτρινο χρώμα στην ίδια εικόνα, προσοχή όμως να ικανοποιούνται οι απαιτήσεις για μεταβλητές και βιβλιοθήκες.

Εικόνα 1: Αρχική οθόνη visual studio code

Εικόνα 2: Αναζήτηση jupyter



Εικόνα 3: Εκτέλεση π

Βιβλιοθήκες:

Pandas:

Η βιβλιοθήκη pandas παρέχει δομές οι οποίες μας διευκολύνουν στη διαχείριση των δεδομένων μας. Από αυτή χρησιμοποιήσαμε κυρίως τα DataFrame. Σε αυτά φορτώσαμε τα δεδομένα που μας δόθηκαν, ώστε να τα επεξεργαστούμε και να εξάγουμε τα ζητούμενα αποτελέσματα. Η βιβλιοθήκη παρέχει επίσης τη συνάρτηση read_csv() μέσω της οποίας φορτώνουμε τα δεδομένα.

Numpy:

Η βιβλιοθήκη numpy χρησιμοποιείται κυρίως για τις μαθηματικές πράξεις (κυρίως στο κομμάτι της γραμμικής άλγεβρας) που προσφέρει. Εμείς τη χρησιμοποιήσαμε για τις τιμές (np.nan) που προσφέρει. Αυτές αναγνωρίζονται από τη προηγούμενη βιβλιοθήκη (pandas) και από πολλές άλλες γνωστές βιβλιοθήκες.

Sklearn:

Πρόκειται για μια βιβλιοθήκη, στην οποία έχουν αναπτυχθεί διάφοροι αλγόριθμοι στο πεδίο της μηχανικής μάθησης. Προσφέρει πολλές δυνατότητες για preprocessing δεδομένων, clustering, classification και άλλα. Από αυτή χρησιμοποιήσαμε την υλοποίηση του DBSCAN, καθώς και τη συνάρτηση NearestNeighbors().

Matplotlib:

Πολύ γνωστή βιβλιοθήκη, με τη βοήθεια της οποίας παράγουμε όλα τα γραφήματα.

Seaborn:

Η βιβλιοθήκη αυτή στηρίζεται πάνω στη Matplotlib και προσφέρει περεταίρω δυνατότητες για την οπτικοποίηση αποτελεσμάτων. Στην εργασία μας χρησιμοποιήσαμε τα correlation διαγράμματα (heatmap που προκύπτουν από το correlation matrix των μεταβλητών μας).

Pytorch:

Η βιβλιοθήκη Pytorch αποτελεί πλαίσιο μηχανικής μάθησης ανοιχτού κώδικα που βασίζεται στη βιβλιοθήκη του Torch, που χρησιμοποιείται για εφαρμογές όπως η όραση στον υπολογιστή και η επεξεργασία φυσικής γλώσσας που αναπτύχθηκε κυρίως από το Meta AI. Είναι δωρεάν λογισμικό ανοιχτού κώδικα που κυκλοφορεί με την τροποποιημένη άδεια BSD. Χρησιμοποιήθηκε για τον παλινδρομητή LSTM.

Gensim:

Η Gensim είναι μια βιβλιοθήκη ανοιχτού κώδικα για μη επιτηρούμενη μοντελοποίηση θεμάτων, ευρετηρίαση εγγράφων, ανάκτηση με ομοιότητα και άλλες λειτουργίες επεξεργασίας φυσικής γλώσσας, χρησιμοποιώντας τη σύγχρονη στατιστική μηχανική μάθηση. Χρησιμοποιήθηκε για την υλοποίηση της τεχνικής word embedding.

NLTK:

Το εργαλείο φυσικής γλώσσας, ή πιο συχνά NLTK, συμπεριλαμβάνει πληθώρα βιβλιοθηκών και προγραμμάτων για συμβολική και στατιστική επεξεργασία φυσικής γλώσσας για αγγλικά γραμμένα στη γλώσσα προγραμματισμού Python. Χρησιμοποιήθηκε για την αφαίρεση των stopwords.

Διαδικασία Υλοποίησης:

Ερώτημα 1:

Αρχικά φορτώσαμε όλα τα δεδομένα μας σε δύο διαφορετικά dataframe (sourcesDf, demandDf). Κάθε γραμμή αυτών περιέχει και μια μέτρηση, για κάποια συγκεκριμένη χρονική στιγμή. Σε αυτές προστέθηκαν δύο ακόμα στήλες, μια που δηλώνει την ώρα στην οποία έγινε η μέτρηση και μια ακόμα που έχει την ημερομηνία. Τα πεδία αυτά είναι χρήσιμα, ώστε να διατηρήσουμε οργανωμένα τα dataframe, να αντικαταστήσουμε τυχόν null τιμές και να κάνουμε την ομαδοποίηση, ώστε να εντοπίσουμε ημέρες outliers. Κατά τη φόρτωση των δεδομένων παρατηρήσαμε ότι υπήρχαν μετρήσεις που έλλειπαν από τα dataframe, ενώ δηλαδή υπήρχαν μετρήσεις ανά 5 λεπτά για κάθε μέρα, για όλο το εικοσιτετράωρο, σε κάποια αρχεία υπήρχαν λιγότερες. Αυτές τις εντοπίζουμε και εισάγουμε γραμμές που έχουν σε όλες τις μεταβλητές null, εκτός από το Time και το Date.

Έπειτα συνεχίσαμε τον καθαρισμό των δεδομένων μας. Πιο συγκεκριμένα παρατηρήσαμε ότι στο sourcesDf υπήρχαν στήλες που αφορούσαν την ίδια μεταβλητή αλλά είχαν διαφορετικό capitalization (Natural gas, Natural Gas και Large hydro, Large Hydro). Αφού βεβαιωθήκαμε ότι δεν υπάρχουν τιμές ταυτόχρονα και στις δύο αυτές στήλες (όταν δηλαδή έχει τιμή μία από τις δύο, τότε η άλλη στήλη έχει null), προχωρήσαμε στην συγχώνευση αυτών. Τέλος αντικαταστήσαμε όλες τις null τιμές με τους μέσους όρους των μετρήσεων των υπόλοιπων ημερών για εκείνη την ώρα.

Για το Α μέρος του ερωτήματος χρησιμοποιήσαμε τα correlation διαγράμματα που γίνανε plot μέσω της βιβλιοθήκης seaborn, ώστε να εντοπίσουμε συσχετίσεις μεταξύ των μεταβλητών μας και έπειτα, μέσω της matplotlib κάναμε plot τις μεταβλητές εκείνες που είχαν τις μεγαλύτερες συσχετίσεις. Εξάγαμε επίσης γραφήματα για τον μέσο όρο των τιμών μιας μεταβλητής, μέσα στο χρονικό διάστημα $24^{ων}$ ωρών. Γράφτηκαν επίσης συναρτήσεις που κάνουν plot συγκεκριμένο αριθμό τυχαίων ημερών για κάποια από τις μεταβλητές. Η παραπάνω ανάλυση έγινε για το sourcesDf και το demandDf, καθώς επίσης και για τον συνδυασμό τους (dailyDemandSourcesDf). Το τελευταίο dataframe περιέχει τις στήλες Time, Date, τη μεταβλητή current_demand από το demandDf (καθώς και τις προβλέψεις) και μια ακόμα στήλη SourcesSum, που είναι το άθροισμα όλων των πηγών ενέργειας για κάθε μέτρηση σε κάθε ημέρα.

Στο Β μέρος εφαρμόσαμε clustering στους μέσους όρους κάθε ημέρας στα τρία αυτά dataframe, χρησιμοποιώντας τον αλγόριθμο DBSCAN, ώστε να εντοπίσουμε τις ημέρες outlier που μας ζητείται. Ως ελάχιστο αριθμό point που μπορούν να χαρακτηριστούν cluster ορίσαμε τη σταθερά 2*(αριθμός μεταβλητών στο dataframe, χωρίς φυσικά τις στήλες Date και Time). Τη σταθερά eps του αλγορίθμου την βρήκαμε μέσω του kdistance plot, δοκιμάζοντας διάφορες τιμές κοντά στο elbow της συνάρτησης.

Στο Γ μέρος πρέπει εκπαιδεύσουμε έναν παλινδρομητή βασισμένο σε LSTM νευρωνικά δίκτυα ο οποίος να μαντεύει για κάθε στιγμή της ημέρας πόση ενέργεια απαιτείται να παραχθεί από μη ανανεώσιμες πηγές. Έτσι αρχικά ορίζουμε το X το οποίο αποτελείται από της έξης στήλες του sourcesDf (Time ,coal ,nuclear,natural gas). Έπειτα ορίζουμε το Y το οποίο είναι η στήλη του demandDf (current_demand). Για να βελτιώσουμε την εκπαίδευση μας κάνουμε κανοικοποίηση στο X και στο Y .Στο X χρησιμοποιούμε standard scaler καθώς είναι αυτός που λειτουργεί καλυτέρα σε δεδομένα με εκτροπές τιμές. Στο Y χρησιμοποιούμε min max scaler καθώς επιθυμούμε να εκπαιδεύσουμε ένα παλινδρομητή αρά τα δεδομένα θα θέλαμε να έχουν τιμές από 0 έως 1. Έπειτα χωρίζουμε τα δεδομένα σε trainset και testset με αναλογία 80/20. Τα δεδομένα μετασχηματίζονται έτσι ώστε να έχουν το κατάλληλο μέγεθός για την είσοδο τους στο δίκτυο LSTM. Στο class LSTM1 ορίζουμε την αρχιτεκτονική των νευρωσικού δικτύου δηλαδή της εποχές , το learning rate ,το μέγεθός εισόδου ,το hidden size,το αριθμό των layers και τον αριθμό των κλάσεων εξόδου .Κατά την εκπαίδευση ελέγχεται το train loss και το validation loss. Όταν το validation loss αρχίσει να αυξάνεται (αν και το train loss μειώνεται) σταματάμε την εκπαίδευση γιατί το δίκτυο από το σημείο αυτό και έπειτα εμφανίζει το φαινόμενο του overfit.

Ερώτημα 2:

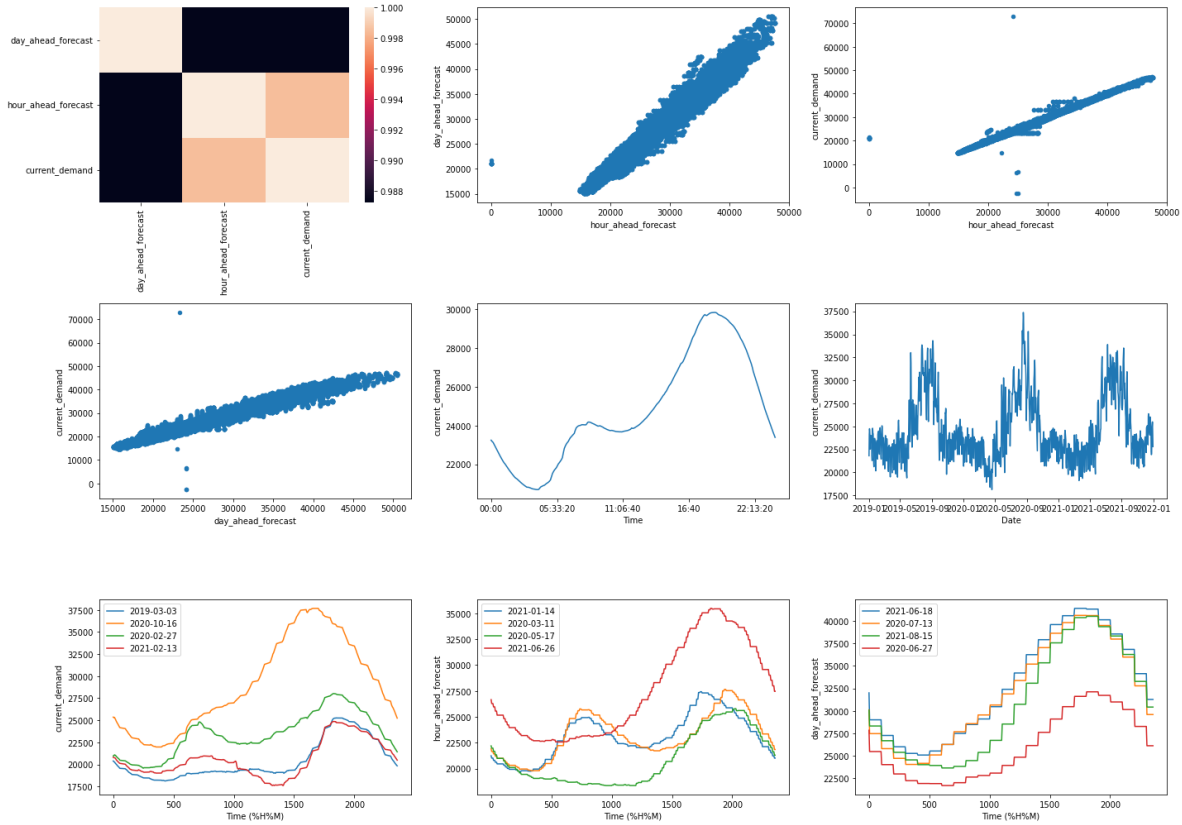
Αρχικά ,φορτώνουμε το csv αρχείο σε ένα dataframe (amazon).Σε πρώτη φάση επεξεργαζόμαστε τα δεδόμενα έτσι ώστε να τα ετοιμάσουμε για την μετατροπή σε διανύσματα χρησιμοποιώντας word embeddings.Αφαιρούμε λοιπόν κάποιες σειρές στις οποίες η κριτική δεν διαθέτει κάποια πληροφορία για παράδειγμα η σειρά με index 6930 οπού το περιεχόμενο του text είναι “:” ενώ παράλληλα αφαιρέσουμε και τα stop words μικρές δηλαδή λέξεις οι οποίες δεν έχουν σημασιολογική προσφορά (and ,the) αλλά και διαφορά σύμβολα. Η τεχνική των word embeddings γίνεται με τη βιβλιοθήκη gensim. Αφού φτιάξουμε και εκπαιδεύσουμε το μοντέλο αθροίζουμε κάθε διάσταση καθεμίας λέξης που εμπεριέχεται στη κριτική. Αυτό γίνεται για να μπορέσουν τα δεδομένα μας να έχουν το ίδιο μέγεθος γεγονός το οποίο είναι απαραίτητο για τον classifier μας ο οποίος είναι ο Random Forest. Χωρίζουμε τα επεξεργασμένα δεδομένα σε training και test dataset και έπειτα τα εισάγουμε στο classifier για να κάνουμε πρόβλεψη των rating scores.

Αποτελέσματα:

Ερώτημα 1:

Μέρος Α:

Όπως φαίνεται στο heatmap (1η γραμμή 1η στήλη) της Εικόνα 4 και οι τρεις μεταβλητές είναι πάρα πολύ συσχετισμένες, το οποίο είναι και το αναμενόμενο, αφού η μια στήλη πρόκειται για την μέτρηση εκείνης της ώρας και οι άλλες δύο είναι προβλέψεις για τη μέτρηση σε μια ώρα και τη μέτρηση της επόμενης μέρας (στο ίδιο timestamp). Στα τρία επόμενα γραφήματα (2 της 1ης γραμμής και το 1ο της 2ης) κάνουμε plot τους διαφορετικούς συνδυασμούς των μεταβλητών, οπότε βλέπουμε ξανά τη μεγάλη συσχέτισή τους. Σε αυτά τα scatter plot μπορούμε να παρατηρήσουμε ότι υπάρχουν και τιμές outlier, εκείνες που ξεφεύγουν από τη γραμμή που χαρακτηρίζει την κάθε γραφική παράσταση. Παρατηρούμε επίσης ότι οι μεταβλητές εκείνες που έχουν μικρότερη συσχέτιση (day_ahead_forecast – hour_ahead_forecast) έχουν μεγαλύτερο spread στον κάθετο άξονα της γραμμής αυτής. Το αμέσως επόμενο γράφημα είναι οι μέσοι όροι των μετρήσεων (current_demand) για όλες τις ημέρες και βλέπουμε ότι είναι πολύ κοντά στη συμπεριφορά κάθε μια από τις ημέρες που φαίνονται στο 1ο γράφημα της 3ης γραμμής. Στο γράφημα (2η γραμμή, 3η στήλη) έχουν γίνει plot όλες οι τιμές του current demand.

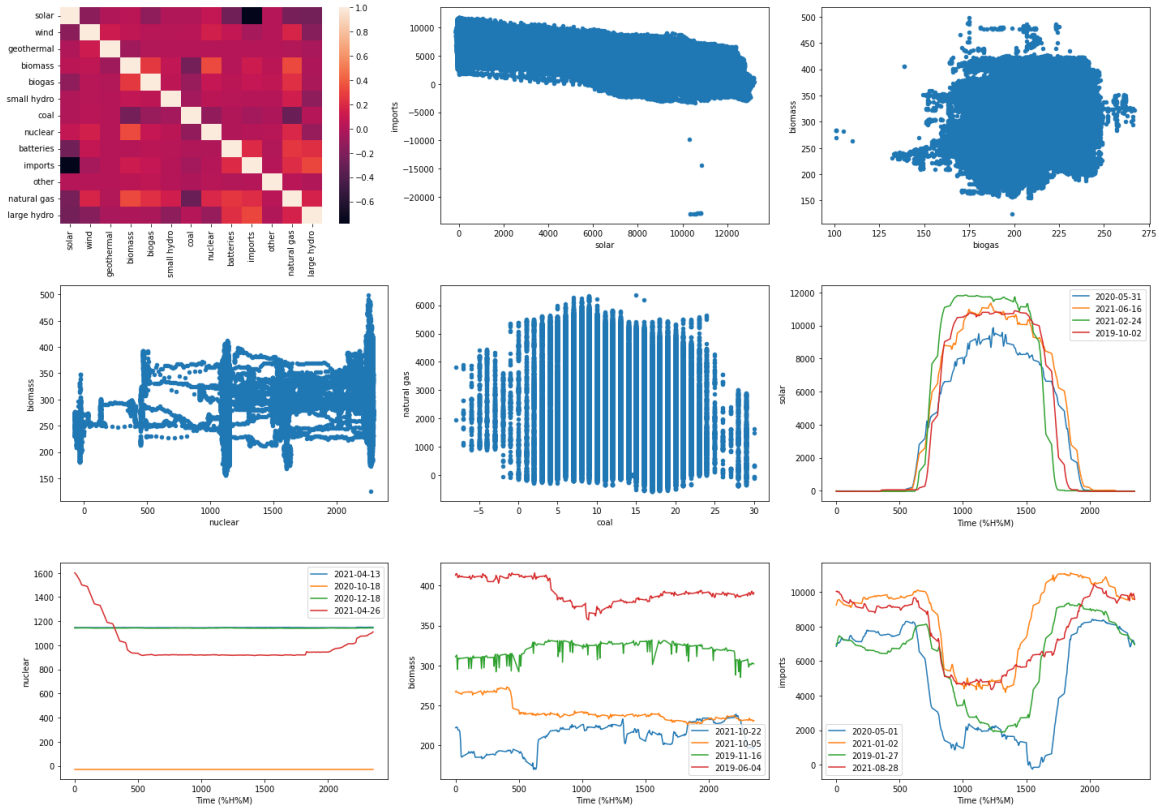


Εικόνα 4: Γραφήματα για το demand, εντοπισμός συσχετίσεων

Πίνακας 1: Στατιστικά μεγέθη demandDf

	count	mean	std	min	25%	50%	75%	max
day_ahead_forecast	315648	24955.46	4950.55	15053	21401	23954	26952.25	50485
hour_ahead_forecast	315648	24839.71	4857.288	0	21369	23889	26827	47697
current_demand	315648	24827.49	4866.348	-2651	21355	23870	26813	72922

Όμοια στην Εικόνα 5 γίνεται αντίστοιχη απεικόνιση των μεταβλητών του dataset. Στο heatmap βλέπουμε πως οι μεταβλητές μας δεν έχουν κάποια ιδιαίτερη συσχέτιση, πέρα από την αντιστρόφως ανάλογη σχέση που φαίνεται να έχουν οι εισαγωγές (imports) με την ηλιακή ενέργεια (solar). Στα επόμενα γραφήματα επιλέξαμε να κάνουμε plot τις μεταβλητές εκείνες που έχουν τη μεγαλύτερη συσχέτιση, με βάση το correlation heatmap. Αυτή τη φορά παρατηρούμε περίεργη συμπεριφορά στα scatter plots την οποία δεν μπορώ να εξηγήσω, παρόλα αυτά και σε εδώ υπάρχουν εμφανείς outlier τιμές. Τέλος παρουσιάζονται γραφήματα από μετρήσεις για κάποια συγκεκριμένη μεταβλητή σε μια τυχαία μέρα.

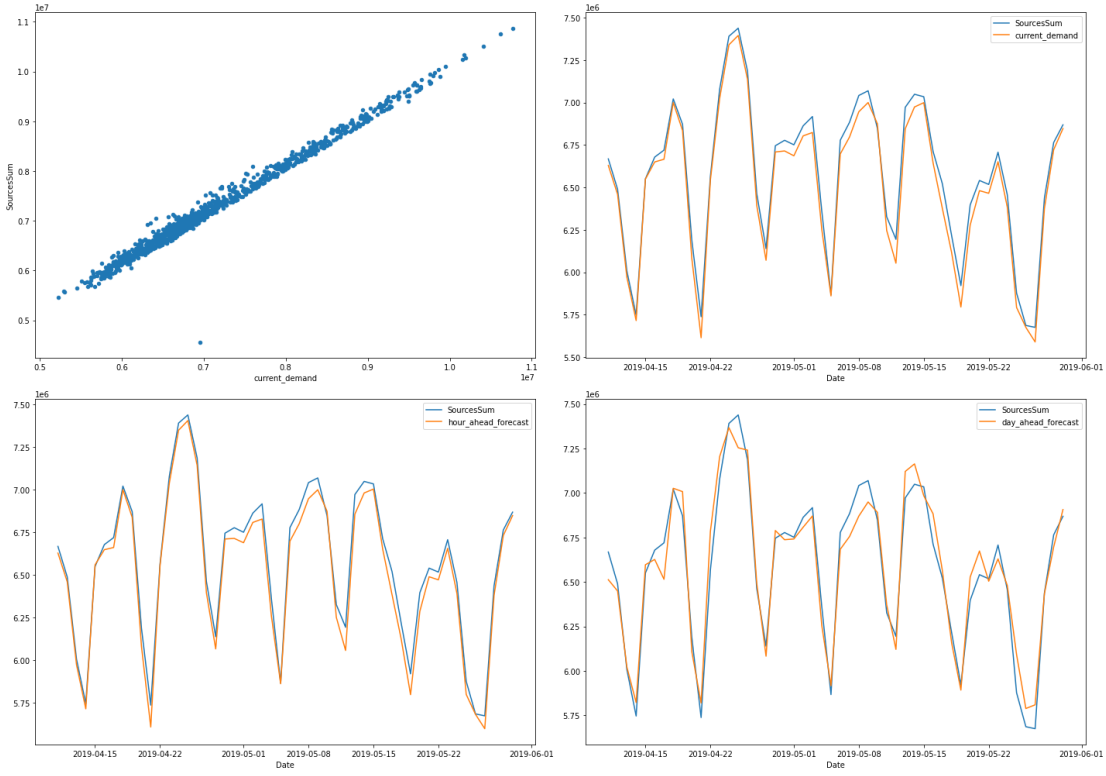


Εικόνα 5: Γραφήματα για τα sources, εντοπισμός συσχετίσεων

Πίνακας 2: Στατιστικά μεγέθη sourcesDf

	count	mean	std	min	25%	50%	75%	max
solar	315648	3559.999	4373.583	-145	-34	114	7969	13191
wind	315648	2021.758	1300.482	-2476	870	1834	3053	36275
geothermal	315648	890.2721	115.9186	-134	866	909	958	1134
biomass	315648	311.2546	48.24063	125	279	315	344	499
biogas	315648	212.5673	15.38187	101	204	213	222	267
small hydro	315648	264.0143	778.0398	-2826	166	227	338	431490
coal	315648	14.24414	4.574259	-8	11	15	18	30
nuclear	315648	1866.838	578.7846	-74	1145	2259	2269	2290
batteries	315648	-5.02324	191.035	-1614	-31	-3	17	1479
imports	315648	6029.531	2614.035	-22949	4250	6467	8069	11832
other	315648	0.001125	0.033517	0	0	0	0	1
natural gas	315648	1897.232	1288.997	-576	908	1551	2733	6362
large hydro	315648	8286.21	3979.844	-280718	5474	7999	10372	26540

Τελευταία ανάλυση του dataset ήταν πάνω στο συνδυασμό τους, όπως αυτός περιγράφεται στη διαδικασία υλοποίησης. Από την Εικόνα 6 φαίνεται η συσχέτιση που υπάρχει μεταξύ των δύο μεταβλητών παραγωγής και ζήτησης. Στις άλλες 3 γραφικές παραστάσεις πως το SourcesSum είναι πάντα πάνω από το current_demand, πράγμα που συμβαίνει και στην περίπτωση της πρόβλεψης για την επόμενη ώρα. Σε ότι αφορά την πρόβλεψη για την επόμενη μέρα παρατηρούμε ότι το προβλεπόμενο demand συμβαίνει να είναι πάνω από τη παραγωγή ενέργειας σε κάποιες ημέρες. Αυτό μπορεί να οφείλεται σε κάποιο error λόγω της φύσης της μεταβλητής (πρόβλεψη).

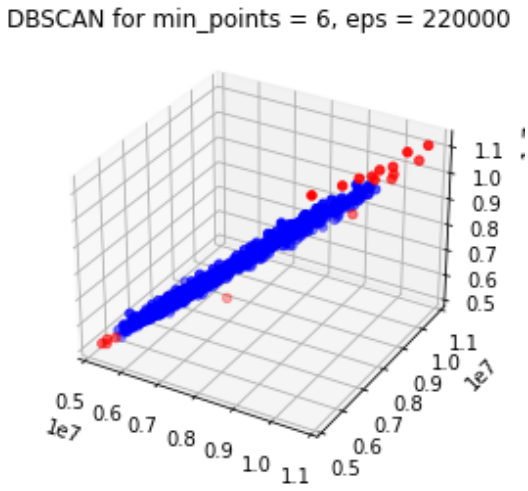


Εικόνα 6: Ανάλυση του dataframe - συνδυασμός demand και sources

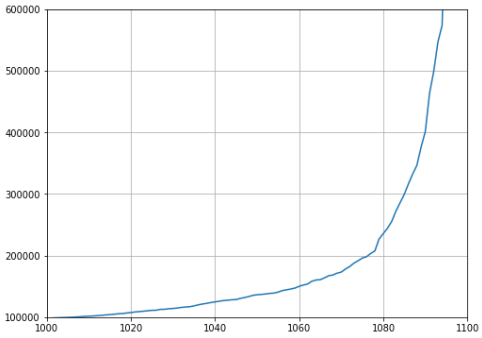
Μέρος Β:

Στα προηγούμενα dataframe εφαρμόσαμε και clustering, χρησιμοποιώντας τον αλγόριθμο DBSCAN. Αυτός μας κατηγοριοποιεί τα δεδομένα σε clusters που ικανοποιούν τις συνθήκες min_points και eps (minimum distance between samples in a cluster). Τα σημεία τα οποία δεν μπορούν να μπουν σε κάποιο cluster, τα σημειώνει ως outlier και τους δίνει το label -1.

Για το demandDf, το clustering θα γίνει στις 3 διαστάσεις (αφού έχουμε 3 μεταβλητές). Προκύπτει το αποτέλεσμα της Εικόνα 7, στο οποίο φαίνεται ότι υπάρχει ένα μοναδικό cluster, με χρώμα μπλε και οι outlier τιμές, με κόκκινο χρώμα (14 σε αριθμό).

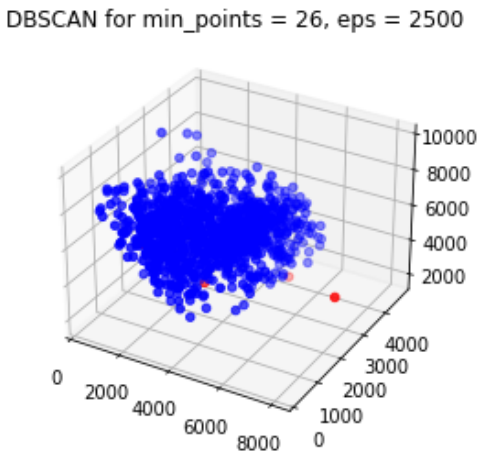


Εικόνα 7: Demand Clustering

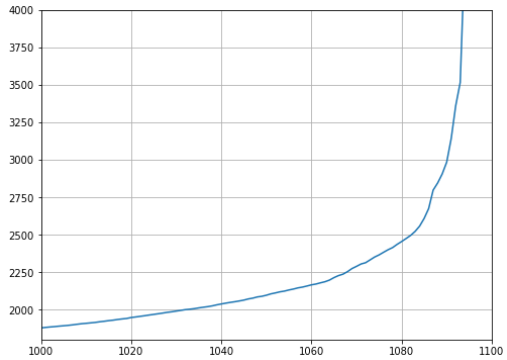


Εικόνα 8: Demand kdistance plot

Το clustering για το sourcesDf, επέστρεψε μικρότερο αριθμό από ημέρες outlier (5). Επειδή είχαμε πολλές μεταβλητές (13), παρουσιάζουμε τα clusters για ένα subset αυτών, solar, wind και import, στην Εικόνα 10.

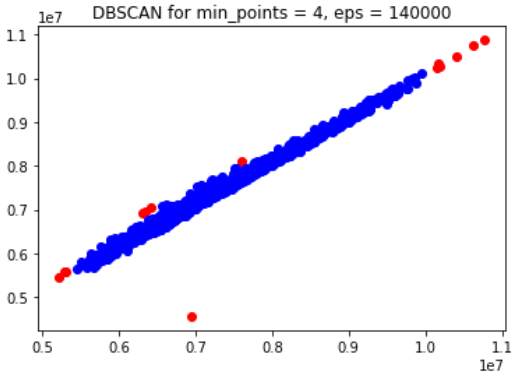


Εικόνα 10: Sources Clustering

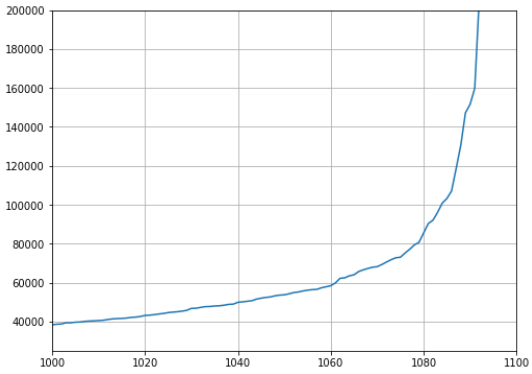


Εικόνα 9: Sources kdistance plot

Τέλος, από το clustering του συνδυασμού των δύο dataframe προέκυψαν 14 ημέρες outlier, όπως φαίνεται στην Εικόνα 12.



Εικόνα 12: Demand - Sources Clustering



Εικόνα 11: Demand - Sources kdisatnce plot

Στους επόμενους πίνακες παραθέτουμε τις ημερομηνίες των outlier μέσων όρων.

Πίνακας 3: Outliers του demandDf

demandDf - outliers
2019-08-18
2020-03-29
2020-04-11
2020-04-12
2020-04-19
2020-08-14
2020-08-15
2020-08-17
2020-08-18
2020-08-19
2020-08-20
2020-08-21
2020-08-22
2020-08-24

sourcesDf - outliers
2019-01-02
2019-10-01
2020-08-18
2021-07-11

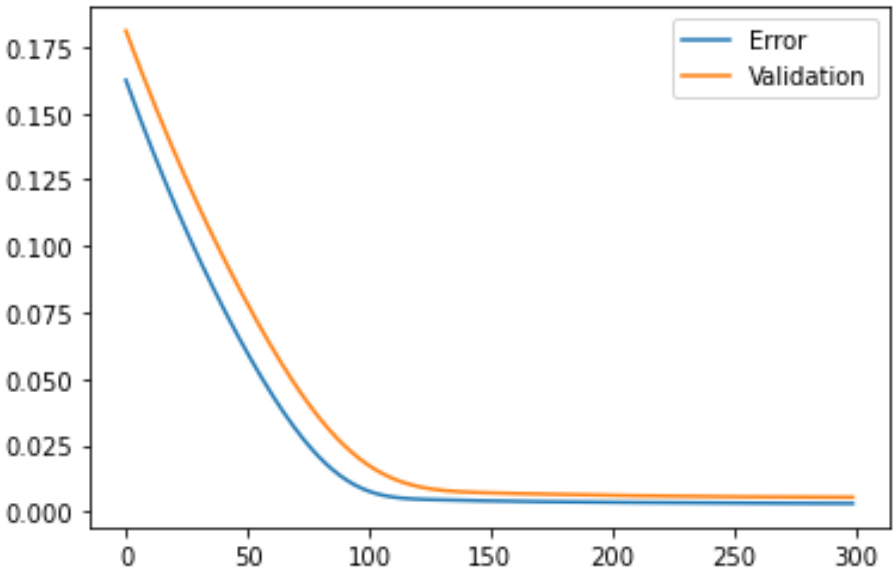
Πίνακας 4: Outliers του sourcesDf

demandSourcesDf - outliers
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
2020-09-06
2021-12-14
2021-12-24
2021-12-25
2021-12-26

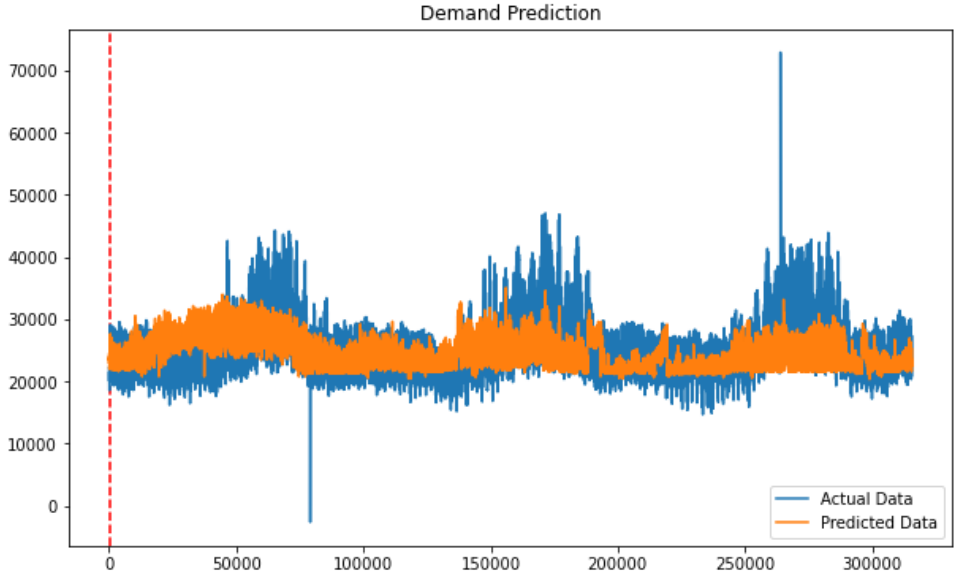
Πίνακας 5: Ημέρες outlier του demandSourcesDf

Μέρος Γ:

Τα αποτελέσματα προέκυψαν για learning rate ίσο με 1e-3/5,input size τρία ,hidden size 64 ,αριθμό layers 1 και αριθμό output κλάσεων 1. Παρακάτω βρίσκονται οι καμπύλες της εκπαίδευσης του δικτύου για 300 εποχές.



Εικόνα 13: Train loss-validation loss



Εικόνα 13: Actual demand-Predicted demand

Παρατηρούμε πως οι predicted τιμές του δικτύου αν και προσεγγίσουν αρκετά τις πραγματικές τιμές, δεν ακολουθούν τις έκτροπες τιμές.

Ερώτημα 2:

Έγιναν διαφορά πειράματα αλλάζοντας τις παραμέτρους των word embeddings αλλά και του random forest. Τα καλυτέρα αποτελέσματα κυμαίνονται στις παρακάτω τιμές .

	precision	recall	f1-score	support
1	0.53	0.17	0.25	1016
2	1.00	0.01	0.02	581
3	0.86	0.02	0.05	793
4	0.42	0.03	0.05	1709
5	0.61	0.99	0.76	5900
accuracy			0.61	9999
macro avg	0.68	0.24	0.23	9999
weighted avg	0.61	0.61	0.49	9999

Εικόνα 15: Απόδοση συστήματος

Η ακρίβεια των προβλεπόμενων τιμών κυμαίνεται από 58% έως 62%.Ενας πιθανός λόγος που δεν πετυχαίνουμε καλυτέρα αποτελέσματα είναι το γεγονός ότι τα δεδομένα μας δεν είναι ισορροπημένα .Συγκεκριμένα ,από τα δεδομένα μας το 60% είναι κριτικές με 5 σκορ ,το 16% με 4 ,το 10% με 1 ,το 8% με 2 και το 6% με 3.