



Εθνικό Μετσόβιο Πολυτεχνείο

Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

ΔΠΜΣ Επιστήμη Δεδομένων και Μηχανική Μάθηση

Ανάλυση και Επεξεργασία Γεωχωρικών Δεδομένων - Σειρά 4η

Σωτήρης Πελέκης

03400069

ΕΔΕΜΜ

25/04/2020

Περιεχόμενα

Εισαγωγή.....	3
Περιγραφή των εφαρμογών.....	3
Εφαρμογή 1.1 - Heatmaps of demand densities	4
Εφαρμογή 1.2 - Timeseries viewer.....	4
Εφαρμογή 1.3	5
Εφαρμογή 1.4	7
Εφαρμογή 2.0	8
Συμπεράσματα.....	10

Εισαγωγή

Στα πλαίσια της άσκησης αυτής δημιουργήθηκε μια σειρά από διαδικτυακές εφαρμογές θέασης μεταφόρτωσης και επεξεργασίας γεωχωρικών δεδομένων. Ο κώδικας και τα αποτελέσματα της άσκησης βρίσκονται στο αποθετήριο github:

<https://github.com/pelekhs/Geo-exercise-4-Sotiris-Pelekis>

Έγινε χρήση των βιβλιοθηκών python:

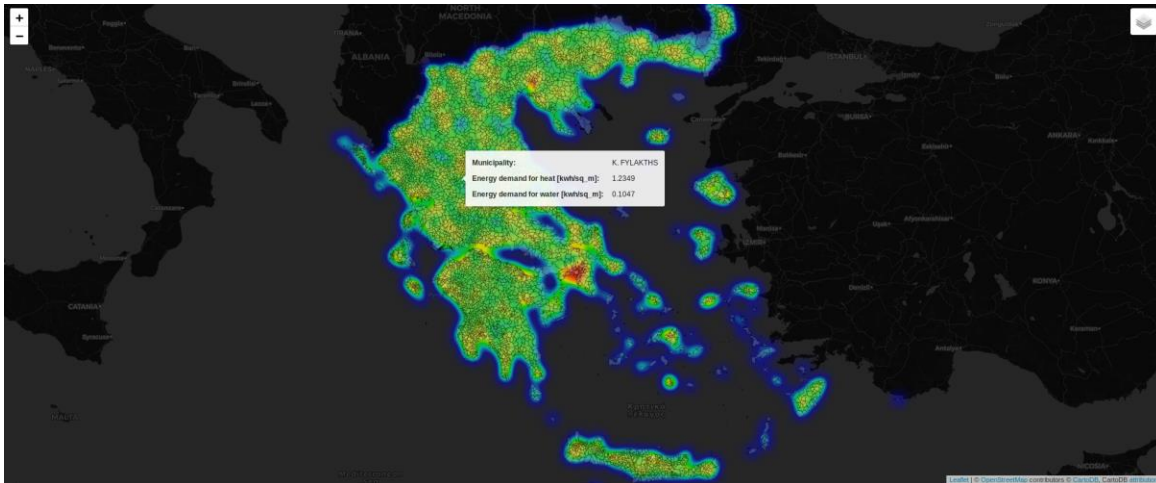
- Folium: Δημιουργία διαδραστικών χαρτών σε notebooks οι οποίοι μπορούν και να εξαχθούν σε .html αρχείο για διαδικτυακή προβολή με την προϋπόθεση ότι είναι διαθέσιμος ο εξυπηρετητής παροχής των γεωχωρικών δεδομένων (Geoserver). (App 1.1, 1.2, 1.3)
- OWSLib: Εκκίνηση επικοινωνίας WMS, WFS, WCS (OGC client) με το geoserver, εξερεύνηση των layers που αυτός παρέχει και των properties τους καθώς και δυνατότητα λήψης αυτών. Έγιναν κάποιοι αρχικοί πειραματισμοί με τη βιβλιοθήκη στο αρχείο Experimentation with OWSlib and WMS-WFS-WCS.ipynb τα προϊόντα του οποίου χρησιμοποιήθηκαν σε τμήματα σε όλες τις εφαρμογές.
- Ipywidgets: Δημιουργία διαδραστικού περιβάλλοντος αναζήτησης εντός jupyter notebook στον κατάλογο τηλεπισκοπικών δεδομένων που κατασκευάστηκε στην προηγούμενη άσκηση για εφαρμογή φίλτρων στα διαθέσιμα δεδομένα του geoserver, για προβολή και λήψη των αποτελεσμάτων μέσω διαδραστικού περιβάλλοντος και χωρίς πληκτρολόγηση κώδικα. (App 2)
- Ipyleaflet: Δημιουργία διαδραστικού προσφέροντας την επιπλέον δυνατότητα σε σχέση με το folium για τροποποίηση των υπάρχοντων layers ενός στημένου χάρτη μετά την εμφάνιση αυτού. Ωστόσο παρουσιάζει το σοβαρό μειονέκτημα ότι δε μπορεί να εξαχθεί σε ένα αρχείο html και να τρέξει χωρίς python kernel στο background. Ωστόσο σε επίπεδο notebooks παρέχει πολύ μεγαλύτερο βαθμό interaction σε σχέση με το folium ειδικά σε συνδυασμό με την ipywidgets. (App 1.4)

Περιγραφή των εφαρμογών

Στην ενότητα αυτή γίνεται μια συνοπτική παρουσίαση κυρίως της λειτουργικότητας εφαρμογών που δημιουργήθηκαν ενώ αναλυτικές λεπτομέρειες σε επίπεδο στησίματος και κώδικα βρίσκονται εντός των συνημμένων notebooks. Δίνονται επίσης κάποια screenshots προκειμένου να φανεί η αλληλεπίδραση με τα δεδομένα καθώς στα εξαχθέντα .html αρχεία να μην φαίνονται οι λειτουργίες, ωστόσο σε ορισμένες περιπτώσεις απουσιάζουν κάποια layers δεδομένων καθώς παρέχονται από τα τοπικά στημένα Geoserver και Postgres. Η εφαρμογή 1.1 και εν μέρει η 1.2 ωστόσο είναι λειτουργικές καθώς λειτουργούν με basemaps οι οποίοι βρίσκονται στο διαδίκτυο ενώ επιπλέον τα vector δεδομένα έχουν κατεβαστεί τοπικά από το geoserver πριν παρουσιαστούν και παρέχονται στον φάκελο των αρχείων. Τα html αρχεία εξόδου των εφαρμογών 1.1-1.3 βρίσκονται στο φάκελο results. Οι άλλες 2 εφαρμογές αναγκαστικά πρέπει να τρέχουν πάνω σε notebook καθώς απαιτούν ipython kernel. Τέλος, διευκρινίζεται ότι οι εφαρμογές 1.1-1.4 απαντούν στο πρώτο ερώτημα της εργασίας ενώ η εφαρμογή 2 στο 2ο.

Εφαρμογή 1.1 - Heatmaps of demand densities

Η εφαρμογή αυτή περιλαμβάνει το παρακάτω διαδραστικό περιβάλλον:



Εικόνα 1: Application 1.1 - Εξερεύνηση της πυκνότητας ζήτησης ενέργειας για θέρμανση και ZNX στους δήμους της Ελλάδας

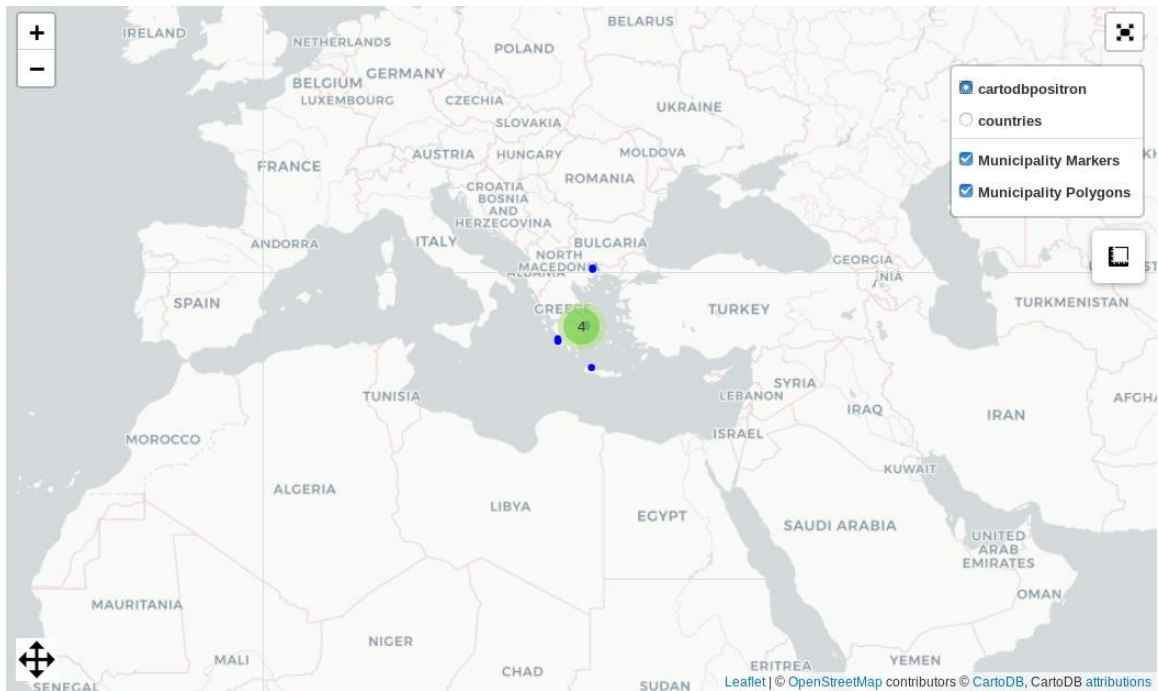
Ο χρήστης μπορεί βλέπει τον χάρτη δήμων της Ελλάδας. Μέσω του widget LayerControl ο χρήστης επιλέγει αν θέλει να βλέπει το heatmap για ZNX ή θέρμανση. Σύμφωνα και με τις προηγούμενες εργασίες βλέπουμε τα πιο θερμά χρώματα μεγάλες τιμές να συγκεντρώνονται στα αστικά κέντρα (π.χ Αθίνα-Θεσσαλονίκη). Αν ο χρήστης θέλει πιο πολλές λεπτομέρεις με hover του ποντικιού πάνω στο πολύγωνο ενός δήμου εμφανίζονται οι πυκνότητες ζήτησης ενέργειας για ZNX και θέρμανση συνοδευόμενες από το όνομα του δήμου. Το hovering effect δημιουργήθηκε με χρήση της παραμέτρου tooltip του GeoJson layer που χρησιμοποιήθηκε στο Folium για την εισαγωγή των vector δεδομένων από τον Geoserver και την Postgres. Στην εφαρμογή αυτή έγινε χρήση τόσο της postgres όσο και του WFS μέσω OWSLib για τροφοδοσία των vector δεδομένων των δημοτικών ενοτήτων στο χάρτη. Ωστόσο λόγω των πολλών δεδομένων το notebook αυτό είναι αρκετά βαρύ.

- ☐ cartodbdark_matter
- ☒ Heat demand density
- ☐ Water demand density
- ☒ Density

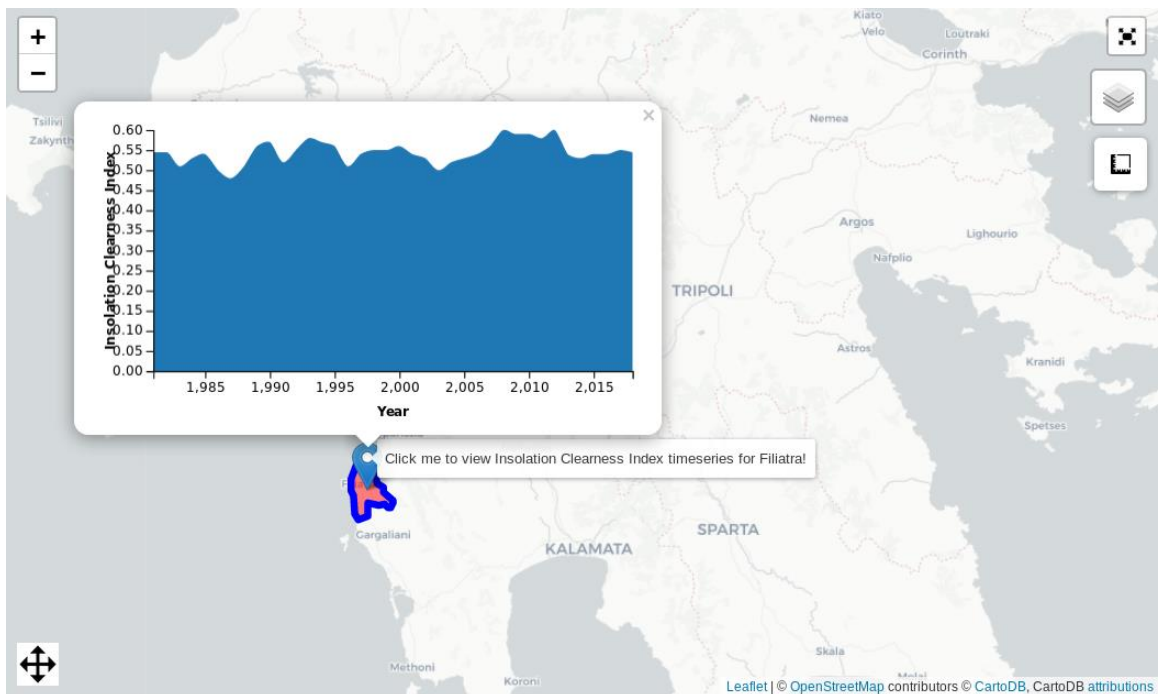
Εικόνα 2: Layer Control Widget

Εφαρμογή 1.2 - Timeseries viewer

Στην εφαρμογή αυτή έγινε χρήση WFS services με την OWSLib για λήψη των γεωμετριών των 4 περιοχών για τις οποίες διαθέτουμε χρονοσειρές. Τα δεδομένα ζητούνται από το geoserver και αυτόματα μέσω της custom συνάρτησης getFeature αποθηκεύονται τοπικά στο relative directory “/wfs”. Στη συνέχεια χρησιμοποιήθηκε το framework Vincent για δημιουργία διαγράμματος χρονοσειρών (insolation clearness index) τα οποία να μπορούν να ενταχθούν σε pop-up πάνω στις πινέζες των 4 επιθυμητών περιοχών. Τέλος έγινε χρήση αρκετών widgets του Folium, όπως full screen, δυνατότητα διεξαγωγής μετρήσεων αποστάσεων (Measure Control) ομαδοποίησης πινεζών (Marker Cluster). Ο χρήστης λοιπόν κατά την περιήγηση του μπορεί να περιηγηθεί στο χάρτη του κόσμου, βλέποντας πινέζες και πολύγωνα στους Δήμους Δράμας, Αθηναίων, Φιλιατρών και Χανίων. Κάνοντας hover στις πινέζες και τα πολυγωνα βλέπει πληροφορίες σχετικές με τον κάθε δήμο, ενώ κλικάροντας σε κάποια πινέζα παίρνει ένα διάγραμμα της διακύμανσης του δείκτη καθαρότητας ηλιακής ακτινοβολίας από το 1981 έως και το 2018.



Εικόνα 3: Εφαρμογή εξερεύνησης περιοχών με διαθέσιμες χρονοσειρές (low zoom scale)



Εικόνα 4: Εφαρμογή εξερεύνησης περιοχών με διαθέσιμες χρονοσειρές (high zoom scale)

Εφαρμογή 1.3

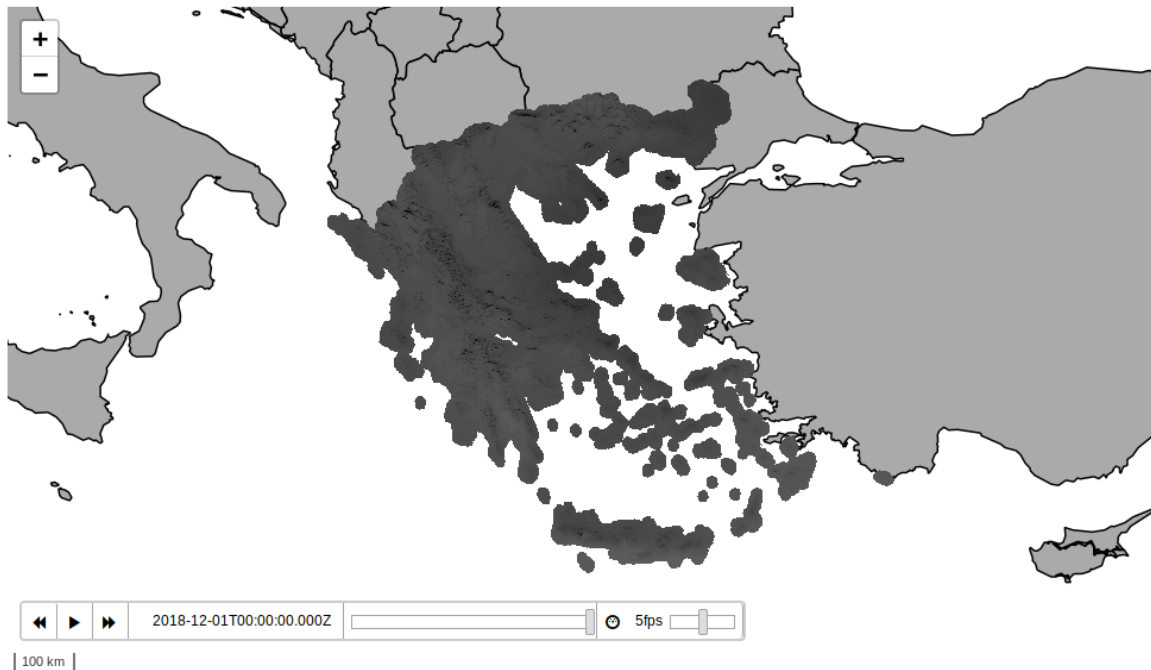
Η εφαρμογή αυτή είναι απλούστερη από τις προηγούμενες και εκμεταλλεύεται την ιδιότητα του WMS να αποθηκεύει Time dimension ενός layer. Έχοντας αποθηκεύσει στο Geoserver τα 12 raster

layers μέσης μηνιαίας εκτίμησης δυναμικού παραγωγής φωτοβολταϊκής παραγωγής ηλεκτρικής ενέργειας σαν ένα ενιαίο Mosaic layer με 12 στοιχεία στη διάσταση του χρόνου χρησιμοποιήθηκε το WMSTileLayer και το plugin TimestampedWmsTileLayers με περίοδο ενός μήνα του Folium μαζί με την υποστήριξη της OWSlib για fetching των metadata του layer από των geoserver (θα μπορούσε εναλλακτικά να είχε γίνει και με χρήση του καταλόγου της προηγούμενης άσκησης) και κατασκευάστηκε η παρακάτω εφαρμογή η οποία δίνει τη δυνατότητα να εξερευνήσει τα διαφορετικά rasters του κάθε μήνα σέρνοντας το διακοπτάκι στο κάτω μέρος. Προφανώς είναι βλέπει κανείς ότι κατά τους χειμερινούς μήνες οι τιμές των pixels είναι εν γένει πιο σκούρες που δηλώνει χαμηλότερο potential φωτοβολταϊκής παραγωγής. Στο background layer τέθηκε το κλασικό layer χωρών του Natural Earth το οποίο βρίσκεται αποθηκευμένο στο Geoserver και πάλι με χρήση WMS μέσω του WMSTileLayer αντί για χρήση κάποιου default base map του folium.

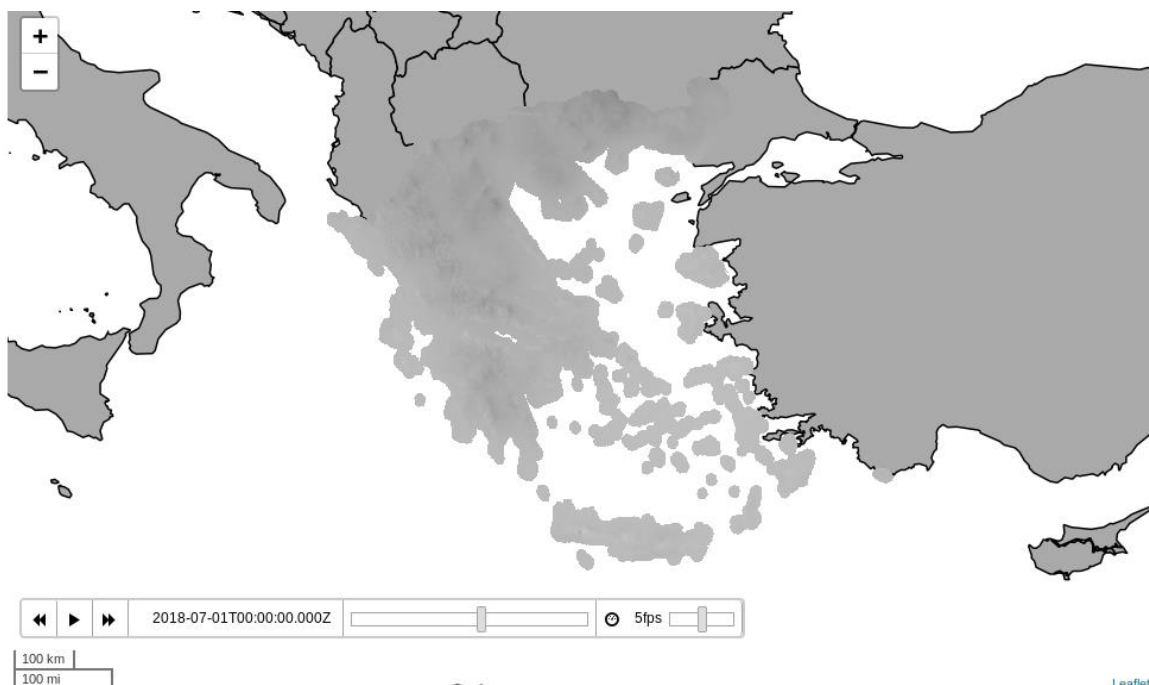
This map consists of one time scrolling layer: PVOUT

Abstract:

Longterm monthly average of potential photovoltaic electricity production (PVOUT) in kWh/kwp, calculated for all months and covering the years from 1994 to 2018



Εικόνα 5: PVOUT time-scrolling application. (Μηνιαίοι μέσοι όροι για το μήνα Δεκέμβριο στα έτη 1994-2018)

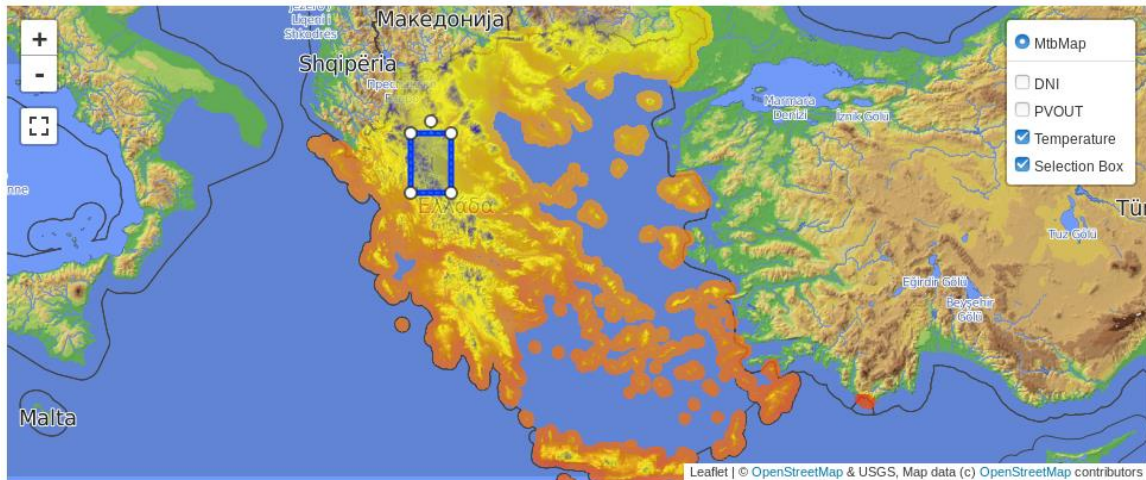


Εικόνα 6: PVOUT time-scrolling application. (Μηνιαίοι μέσοι όροι για το μήνα Ιούλιο στα έτη 1994-2018)

Εφαρμογή 1.4

Η εφαρμογή αυτή είναι ίσως η πιο χρήσιμη εφαρμογή του πακέτου του ερωτήματος 1. Έχει υλοποιηθεί με χρήση της βιβλιοθήκης ipyleaflet η οποία δίνει την επιπλέον δυνατότητα σε σχέση με το folium για τροποποίηση των υπάρχοντων layers ενός στημένου χάρτη μετά την εμφάνιση αυτού με διαδραστικό τρόπο. Στη συγκεκριμένη εφαρμογή παρουσιάζονται 3 raster layers του Geoserver (Μέση ετήσια δυνητική παραγωγή φωτοβολταϊκής ενέργειας – θερμοκρασία – δείκτης κατακόρυφης συνιστώσας ηλιακής ακτινοβολίας (1994-2018)). Ο χρήστης έχει τη δυνατότητα να περιηγηθεί στο χάρτη, επιλέγοντας ποιο από τα 3 Layers επιθυμεί να προβάλει. Τα styles φορτώθηκαν επίσης από το geoserver. Στη συνέχεια του παρέχεται ένα τετραγωνικό layer του οποίου τροποποιήσει διαδραστικά τις συντεταγμένες του πάνω στο σχεδιασμένο χάρτη με χρήση του ποντικιού και όχι με αλλαγή του κώδικα. Στη συνέχεια εκτέλεσης του notebook, με αίτημα WCS στο geoserver (συνάρτηση wcs download() με τη χρήση OWSlib), η εφαρμογή προχωράει σε κατέβασμα των raster δεδομένων και των 3 layers για το bounding box που επέλεξε πάνω στο χάρτη ο χρήστης και προβαίνει σε επεξεργασία με χρήση της βιβλιοθήκης rasterio και numpy ώστε να εξάγει στατιστικά [μέση τιμή φωτοβολταϊκής παραγωγής, θερμοκρασίας και ηλιακής ακτινοβολίας στο επιλεγθέν bounding box (Εικόνα 8)]. Μια τέτοιου τύπου εφαρμογή αυτή θα μπορούσε κάλλιστα να χρησιμοποιηθεί σαν ένα μικρό decision support tool για επενδυτές φωτοβολταϊκών εγκαταστάσεων οι οποίοι επιθυμούν να ενημερωθούν για την ποιότητα των κλιματικών συνθηκών ανα περιοχή προκειμένου να προβούν σε αγορά σχετικών εκτάσεων.

Please move polygon in order to select area for advanced aggregated statistics:



Εικόνα 7: Η εφαρμογή επιλογής πολυγώνου για επεξεργασία των raster δεδομένων που εμπίπτουν στο bounding box

Mean Temperature

```
temp_mean = temp_band[~np.isnan(temp_band)]  
print ("The mean temperature in the selected area is {:.3f} Celsius degrees".format(np.mean(temp_mean)))
```

The mean temperature in the selected area is 10.237 Celsius degrees

Mean Potential Photovoltaic production

```
pvout_mean = pvout_band[~np.isnan(pvout_band)]  
print ("The mean potential photovoltaic production in the selected area is {:.3f} kWh/kWp".format(np.mean(pvout_mean)))
```

The mean potential photovoltaic production in the selected area is 1491.729 kWh/kWp

Mean Direct Normal Irradiation

```
dni_mean = dni_band[~np.isnan(dni_band)]  
print ("The mean direct normal irradiation in the selected area is {:.3f} kWh/m^2".format(np.mean(dni_mean)))
```

The mean direct normal irradiation in the selected area is 1575.812 kWh/m²

Εικόνα 8: Τα αποτελέσματα της επεξεργασίας των raster δεδομένων επιλογής του χρήστη

Εφαρμογή 2.0

Η εφαρμογή αυτή αποτελεί ένα διαδραστικό ευρετήριο πάνω στο κατάλογο τηλεπισκοπικών δεδομένων. Παρέχονται διαδραστικά φίλτρα στο χρήστη για την υλοποίηση γεωχωρικών ερωτημάτων επι του καταλόγου ο οποίος φορτώνεται από την postgres σε μορφή geopandas. Ο χρήστης μπορεί να δει ποια layers του καταλόγου ανταποκρίνονται στις προτιμήσεις του χωρίς να τροποποιήσει ούτε μία γραμμή κώδικα. Αυτό επιτυγχάνεται με χρήση των διαδραστικών widgets που παρέχονται από τη βιβλιοθήκη ipywidgets. Επιπλέον τα αποτελέσματα του query εμφανίζονται live κατά τη χρήση των widgets. Ο χρήστης δηλαδή δε χρειάζεται να ξαναεκτελέσει το cell με περιεχόμενο “display(output)” προκειμένου να δει τα layers που ικανοποιούν τα φίλτρα του.

Check if you want the layer to contain timeseries:

☒ Timeseries only

Choose bounding box to be included in the layer:

'Xmin'

23.0000

'Ymin'

37.0000

'Xmax'

24.0000

'Ymax'

38.0000

Layer Type: raster

Region: Greece

Choose date range to be included in the layer:

From 01/01/1995

To 01/01/2000

Below we get a live update of the above query results:

display(output)

['PVOUT']

Check if you want the layer to contain timeseries:

☒ Timeseries only

Choose bounding box to be included in the layer:

'Xmin'

23.0000

'Ymin'

37.0000

'Xmax'

'Ymax'

24.0000

38.0000

Layer Type: raster

Region: Greece

Choose date range to be included in the layer:

From 01/01/1991

To 01/01/2000

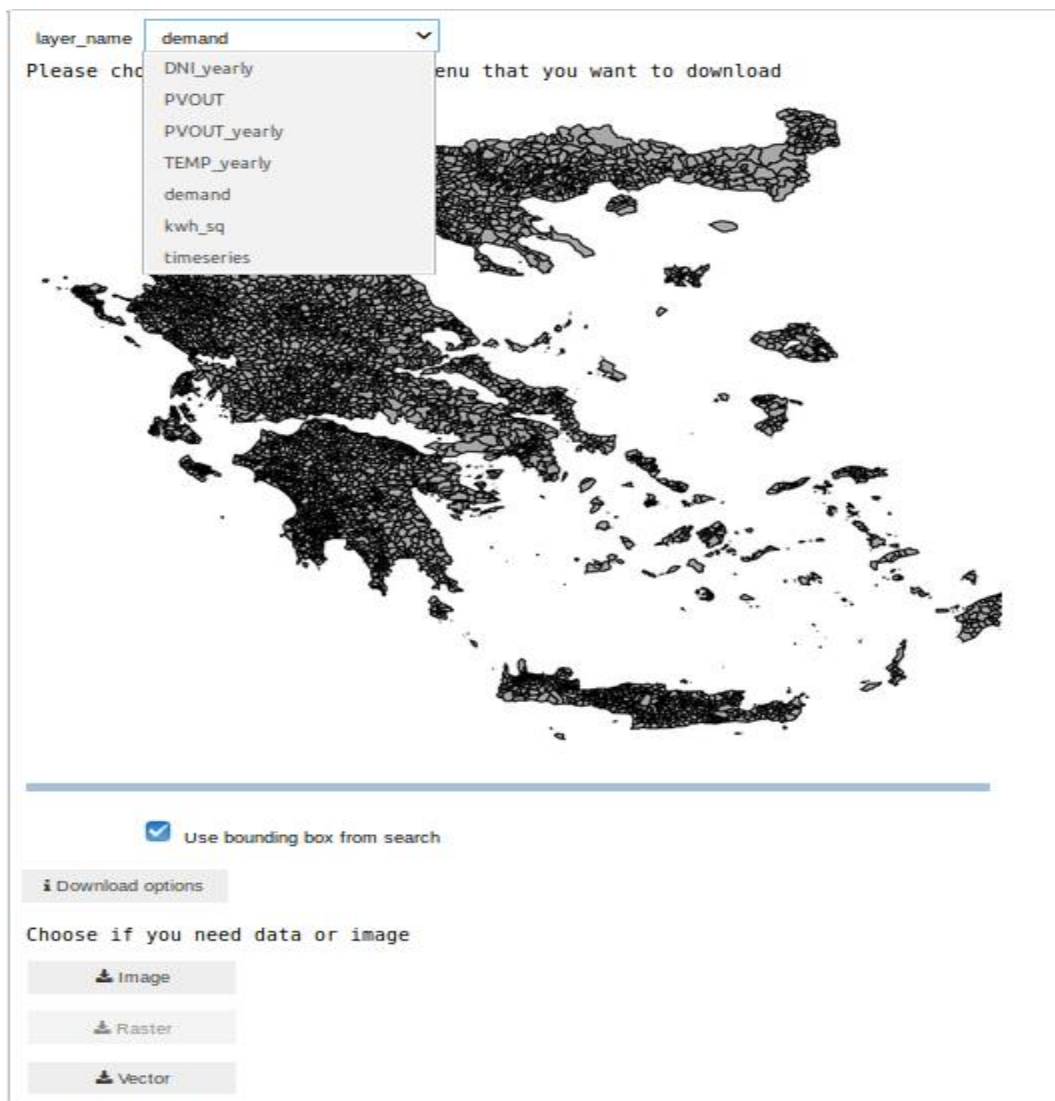
Below we get a live update of the above query results:

display(output)

['timeseries']

Εικόνα 9: Η διεπαφή χρήσης της μηχανής αναζήτησης τηλεπισκοπικών δεδομένων πάνω στον κατάλογο μεταδεδομένων

Στη συνέχεια με επιπλέον χρήση της OWSLib ο χρήστης μπορεί να κάνει προβολή, με χρήση του WMS link του καταλόγου των layers-προϊόντων που ικανοποιούν τα φίλτρα της αναζήτησης του με χρήση ενός dropdown menu. Τέλος, όταν κατασταλάξει στην εικόνα που επιθυμεί να κατεβάσει, μπορεί να το κάνει, επιλέγοντας μάλιστα αν θέλει το layer σε μορφή εικόνας (WMS) ή vector-raster ανάλογα με τον τύπο του layer. Το GUI του ενεργοποιεί αυτόματα μόνο τις επιλογές που είναι διαθέσιμες συμβουλευόμενο ανα πάσα στιγμή τα metadata του καταλόγου ανάλογα με την επιλογή του dropdown. Επίσης, του δίνει τη δυνατότητα είτε να κατεβάσει ολόκληρο το layer είτε μόνο το bounding box που επέλεξε στα φίλτρα. Οι δυνατότητες αυτές του μενού μεταφόρτωσης φαίνονται στην εικόνα 10. Βλέπουμε ότι για το συγκεκριμένο layer της προεπισκόπησης, εφόσον είναι vector layer, είναι ενεργοποιημένο μόνο τα buttons image (WMS) και vector (WFS) τα οποία στο πάτημα καλούν τις συνάρτησεις getMap και getFeature αντίστοιχα οι οποίες αποθηκεύουν κατευθείαν το μεταφορτωμένο αρχείο από το geoserver στους φακέλους wms, wfs αντίστοιχα. Στην περίπτωση που το επιλεγθέν layer είναι τύπου raster ενεργοποιείται το αντίστοιχο button το οποίο καλεί στο πάτημα την wcs download και μας παραπέμπει στο download link του geoserver.



Εικόνα 10: Διεπαφή προεπισκόπησης και μεταφόρτωσης εντός jupyter notebook

Συμπεράσματα

Στα πλαίσια της εργασίας αυτής έγινε πειραματισμός με ποικίλα frameworks python για την κατασκευή διαδραστικών χαρτών. Κατασκευάστηκαν 5 μικροεφαρμογές θέασης αναζήτησης, επεξεργασίας και μεταφόρτωσης γεωχωρικών δεδομένων οι οποίες ωστόσο είναι απαραίτητο να έχουν πρόσβαση στα δεδομένα της Postgres και του Geoserver για την ολοκληρωμένη λειτουργία τους είτε σε περιβάλλον Browser (1.1, 1.2, 1.3) είτε σε περιβάλλον Notebook (1.4, 2.0). Ειδικά οι 2 τελευταίες εφαρμογές παρουσιάζουν ιδιαίτερο χρηστικό ενδιαφέρον καθώς θα μπορούσαν, όπως αναφέρθηκε, να χρησιμοποιηθούν σε πραγματικά use cases

1. Αναζήτηση κατάλληλης τοποθεσίας φωτοβολταϊκών επενδύσεων
2. Διαδραστική αναζήτηση δεδομένων καταλόγου (pandas ή SQL) μέσω ipython notebook αλλά και δεδομένων γεωχωρικού εξηρητητή με δυνατότητες μεταφόρτωσης.