ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΑΝΑΓΝΩΡΙΣΗ ΠΡΟΤΥΠΩΝ

ΑΠΑΛΛΑΚΤΙΚΗ ΕΡΓΑΣΙΑ

ΕΛΕΝΗ ΚΑΤΣΙΜΙΧΑ Π21053

ΓΙΩΡΓΟΣ ΚΟΝΤΟΓΙΩΡΓΟΣ Π21065

ΠΕΙΡΑΙΑΣ

Φεβρουάριος 2024

Περιεχόμενα

Προ-επεξεργασία δεδομένων

Υποσύνολα αριθμητικών και κατηγορικών δεδομένων

Για να αναγνωριστούν τα υποσύνολα των αριθμητικών και κατηγορικών δεδομένων στο αρχικό dataset γίνεται χρήση δυο συναρτήσεων που δέχονται ως ορίσματα τα δεδομένα σε μορφή `pandas.DataFrame` και επιστρέφουν σε λίστα τα αριθμητικά και κατηγορικά χαρακτηριστικά αντίστοιχα.

1. Για την αναγνώριση του υποσυνόλου αριθμητικών δεδομένων του αρχικού dataset χρησιμοποιείται η συνάρτηση `get_numerical_features`. Για την επιλογή των χαρακτηριστικών με τιμές ακέραιους και/ή αριθμούς κινητής υποδιαστολής αξιοποιείται η μέθοδος `select_dtypes` με όρισμα `include='number'`.
2. Για την αναγνώριση του υποσυνόλου κατηγορικών δεδομένων του αρχικού dataset χρησιμοποιείται η συνάρτηση `get_categorical_features`. Για την επιλογή των κατηγορικών χαρακτηριστικών αξιοποιείται η μέθοδος `select_dtypes` με όρισμα `include='object'`.

```
PS C:\Users\helen\source\repos\Noudi03\PatternRecognition> python -m src

Show numerical and categorical features:

The numerical features are:  ['longitude', 'latitude', 'housing_median_age',
 'total_rooms', 'total_bedrooms', 'population', 'households', 'median_income
', 'median_house_value']
The categorical_features are:  ['ocean_proximity']
```

Εικόνα 1: Αναγνώριση αριθμητικών και κατηγορικών χαρακτηριστικών

Έλεγχος αριθμητικών χαρακτηριστικών για ενδεχόμενες ελλιπείς τιμές

Ο έλεγχος των αριθμητικών χαρακτηριστικών για ενδεχόμενες ελλιπείς τιμές επιτυγχάνεται με την συνάρτηση `check_empty_fields`. Η `check_empty_fields` δέχεται 1 παράμετρο:

1. `file_name(str)`: Το όνομα του αρχείου.

Διαδικασία:

1. Χρησιμοποιείται η συνάρτηση `get_numerical_features` για να επιλεχθούν τα αριθμητικά στοιχεία του `DataFrame`.
2. Ελέγχεται εάν υπάρχουν κενά πεδία μεταξύ των αριθμητικών χαρακτηριστικών. Δύο είναι οι περιπτώσεις αυτού του ελέγχου:
   i. Βρέθηκαν κενά πεδία. Τότε εμφανίζεται το πλήθος τους και καλείται η συνάρτηση `fill_empty_fields`.
   ii. Δεν βρέθηκαν κενά πεδία. Σε αυτήν την περίπτωση εμφανίζεται κατάλληλο μήνυμα που μας ενημερώνει ότι δεν υπάρχουν κενά πεδία.

Η συνάρτηση `fill_empty_fields` γεμίζει τα αριθμητικά πεδία του `DataFrame` με την διάμεση τιμή της εκάστοτε αριθμητικής στήλης. Δέχεται 2 παραμέτρους:

1.`df(pandas.DataFrame)`: Το αρχικό `DataFrame`.

2.`path(str)`: Το path στο οποίο θα αποθηκευτεί το τελικό CSV αρχείο.

Διαδικασία:

1. Η συνάρτηση δημιουργεί ένα αντίγραφο του αρχικού `DataFrame`.
2. Συμπληρώνει τα κενά πεδία .
3. Αποθηκεύει το συμπληρωμένο `DataFrame` σε ένα νέο αρχείο CSV.
4. Εμφανίζει τον αριθμό των κενών πεδίων μετά την συμπλήρωση των κενών ο οποίος πρέπει να είναι 0.

```
Check for empty fields:

207 empty fields found in the CSV file.
Missing values after filling empty fields 0
```

Εικόνα 2: Έλεγχος για κενά αριθμητικά πεδία

Κλιμάκωση (Scaling) των αριθμητικών δεδομένων

Για την κλιμάκωση των δεδομένων υλοποιήθηκε η συνάρτηση `scale_data`. Δέχεται 4 παραμέτρους:

1. `df (pandas.DataFrame)`: αντιπροσωπεύει το σύνολο δεδομένων που θα κλιμακωθεί.
2. scaler: Χρησιμοποιούνται οι scalers της βιβλιοθήκης `scikit-learn`:
   i. `sklearn.preprocessing.StandardScaler()`
   ii. `sklearn.preprocessing.MinMaxScaler()`
   iii. `sklearn.preprocessing.RobustScaler()`
3. `scaler_name(str)`: το όνομα του scaler που χρησιμοποιείται για την ονομασία αρχείων και για console outputs.
4. `check_std (bool, Optional)`: Εάν η τιμή του είναι `True`, ελέγχεται η τυπική απόκλιση κάθε αριθμητικής στήλης για τυχών μηδενικές αποκλίσεις.

Διαδικασία:

1. Κλήση της συνάρτησης `get_numerical_features` για να επιλεχτούν τα αριθμητικά στοιχεία
2. Έλεγχος τυπικής απόκλισης: Στην περίπτωση που η τιμή `check_std` είναι True, για κάθε αριθμητικό χαρακτηριστικό ελέγχεται η τυπική απόκλιση. Αν η τιμή είναι 0, αφαιρείται η εκάστοτε στήλη ώστε να μην υπάρξει πρόβλημα διαίρεσης με μηδενικό στοιχείο. Παρόλο που δεν αντιμετωπίζουμε πραγματικά αυτή την πρόκληση, αν υπήρχαν τιμές με μηδενική τυπική απόκλιση, θα προσεγγίζαμε την κατάσταση διαφορετικά.
3. Εφαρμογή Scaler: χρήση του επιλεγμένου scaler και μετατροπή του αποτελέσματος από `NumPy array` σε `DataFrame`.
4. Αποθήκευση και Εκτύπωση:
   i. Δημιουργία νέου αρχείου CSV με βάση τον τύπο του scaler.
   ii. Αποθήκευση σε νέο αρχείο CSV.
   iii. Εκτύπωση του αποτελέσματος.

```
Scale the dataset with Standard Scaler:

The standard deviation of longitude is: 2.0035317235025882
The standard deviation of latitude is: 2.1359523974571153
The standard deviation of housing_median_age is: 12.58555761211165
The standard deviation of total_rooms is: 2181.615251582795
The standard deviation of total_bedrooms is: 419.3918779216883
The standard deviation of population is: 1132.462121765341
The standard deviation of households is: 382.32975283161073
The standard deviation of median_income is: 1.8998217179452688
The standard deviation of median_house_value is: 115395.61587441387
```

Εικόνα 3: Έλεγχος τυπικής απόκλισης

The dataset has been scaled using StandardScaler:

	longitude	latitude	housing_median_age	total_rooms	total_bedrooms	population	households	median_income	median_house_value
0	-1.327835	1.052548	0.982143	-0.804819	-0.972476	-0.974429	-0.977033	2.344766	2.129631
1	-1.322844	1.043185	-0.607019	2.045890	1.357143	0.861439	1.669961	2.332238	1.314156
2	-1.332827	1.038503	1.856182	-0.535746	-0.827024	-0.820777	-0.843637	1.782699	1.258693
3	-1.337818	1.038503	1.856182	-0.624215	-0.719723	-0.766028	-0.733781	0.932968	1.165100
4	-1.337818	1.038503	1.856182	-0.462404	-0.612423	-0.759847	-0.629157	-0.012881	1.172900
...	...	...	...	...	...	...	...	...	...
20635	-0.758826	1.801647	-0.289187	-0.444985	-0.388283	-0.512592	-0.443449	-1.216128	-1.115804
20636	-0.818722	1.806329	-0.845393	-0.888704	-0.922403	-0.944405	-1.008420	-0.691593	-1.124470
20637	-0.823713	1.778237	-0.924851	-0.174995	-0.123608	-0.369537	-0.174042	-1.142593	-0.992746
20638	-0.873626	1.778237	-0.845393	-0.355600	-0.304827	-0.604429	-0.393753	-1.054583	-1.058608
20639	-0.833696	1.750146	-1.004309	0.068408	0.188757	-0.033977	0.079672	-0.780129	-1.017878

[20640 rows x 9 columns]

Εικόνα 4: Κλιμάκωση των αριθμητικών δεδομένων με την χρήση του Standard Scaler

The dataset has been scaled using MinMaxScaler:

	longitude	latitude	housing_median_age	total_rooms	total_bedrooms	population	households	median_income	median_house_value
0	0.211155	0.567481	0.784314	0.022331	0.019863	0.008941	0.020556	0.539668	0.902266
1	0.212151	0.565356	0.392157	0.180503	0.171477	0.067210	0.186976	0.538027	0.708247
2	0.210159	0.564293	1.000000	0.037260	0.029330	0.013818	0.028943	0.466028	0.695051
3	0.209163	0.564293	1.000000	0.032352	0.036313	0.015555	0.035849	0.354699	0.672783
4	0.209163	0.564293	1.000000	0.041330	0.043296	0.015752	0.042427	0.230776	0.674638
...	...	...	...	...	...	...	...	...	...
20635	0.324701	0.737513	0.470588	0.042296	0.057883	0.023599	0.054103	0.073130	0.130105
20636	0.312749	0.738576	0.333333	0.017676	0.023122	0.009894	0.018582	0.141853	0.128043
20637	0.311753	0.732200	0.313725	0.057277	0.075109	0.028140	0.071041	0.082764	0.159383
20638	0.301793	0.732200	0.333333	0.047256	0.063315	0.020684	0.057227	0.094295	0.143713
20639	0.309761	0.725824	0.294118	0.070782	0.095438	0.038790	0.086992	0.130253	0.153403

[20640 rows x 9 columns]

Εικόνα 5: Κλιμάκωση των αριθμητικών δεδομένων με την χρήση του MinMax Scaler

```
The dataset has been scaled using RobustScaler:

       longitude  latitude  housing_median_age  total_rooms  total_bedrooms  population  households  median_income  median_house_value
0      -0.986807  0.957672            0.631579    -0.733422       -0.883755   -0.899787   -0.870769       2.197582            1.880448
1      -0.984169  0.952381           -0.421053     2.924276        1.937906    1.316631    2.243077       2.186664            1.232041
2      -0.989446  0.949735            1.210526    -0.388178       -0.707581   -0.714286   -0.713846       1.707732            1.187941
3      -0.992084  0.949735            1.210526    -0.501691       -0.577617   -0.648188   -0.584615       0.967177            1.113523
4      -0.992084  0.949735            1.210526    -0.294074       -0.447653   -0.640725   -0.461538       0.142854            1.119724
...          ...       ...                 ...          ...             ...         ...         ...            ...                 ...
20635  -0.686016  1.380952           -0.210526    -0.271725       -0.176173   -0.342217   -0.243077      -0.905796           -0.700086
20636  -0.717678  1.383598           -0.578947    -0.841053       -0.823105   -0.863539   -0.907692      -0.448655           -0.706977
20637  -0.720317  1.367725           -0.631579     0.074695        0.144404   -0.169510    0.073846      -0.841709           -0.602239
20638  -0.746702  1.367725           -0.578947    -0.157036       -0.075090   -0.453092   -0.184615      -0.765007           -0.654608
20639  -0.725594  1.351852           -0.684211     0.387002        0.522744    0.235608    0.372308      -0.525816           -0.622222

[20640 rows x 9 columns]
```

Εικόνα 6 :Κλιμάκωση των αριθμητικών δεδομένων με την χρήση του Robust Scaler

Κωδικοποίηση των κατηγορικών δεδομένων

Για την one-hot κωδικοποίηση των κατηγορικών δεδομένων χρησιμοποιείται η συνάρτηση `one_hot_encode_data`, η οποία δέχεται 1 παράμετρο:

1. `df (pandas.DataFrame)`: Το dataset που θα κωδικοποιηθεί.

Διαδικασία:

1. Αρχικοποίηση του one-hot encoder.
2. Επιλογή των κατηγορικών χαρακτηριστικών μέσω της συνάρτησης `get_categorical_features`.
3. Προσαρμογή και μετασχηματισμός των δεδομένων.
4. Μετατροπή του αποτελέσματος σε `NumPy` array.
5. Δημιουργία `DataFrame` από το αποτέλεσμα και console output.

Αποτελέσματα:

Το σύνολο δεδομένων μετά την one-hot κωδικοποίηση σε μορφή binary matrix, όπου κάθε γραμμή αντιπροσωπεύει ένα δείγμα και κάθε στήλη μια κατηγορία του χαρακτηριστικού `'ocean_proximity'`.

Χρησιμοποιείται επίσης η συνάρτηση `append_categorical_data` με την οποία συνδέεται το one-hot κωδικοποιημένο `DataFrame` με το `DataFrame` το οποίο έχει συμπληρωμένες τις ενδιάμεσες τιμές μέσω της μεθόδου `pandas.concat` και το αποτέλεσμα αποθηκεύεται σε ένα CSV αρχείο.

```
One hot encode the categorical features:

The categorical features of the dataset have been one-hot encoded:

       ocean_proximity_<1H OCEAN  ocean_proximity_INLAND  ocean_proximity_ISLAND  ocean_proximity_NEAR BAY  ocean_proximity_NEAR OCEAN
0                              0                       0                       0                         1                           0
1                              0                       0                       0                         1                           0
2                              0                       0                       0                         1                           0
3                              0                       0                       0                         1                           0
4                              0                       0                       0                         1                           0
...                          ...                     ...                     ...                       ...                         ...
20635                          0                       1                       0                         0                           0
20636                          0                       1                       0                         0                           0
20637                          0                       1                       0                         0                           0
20638                          0                       1                       0                         0                           0
20639                          0                       1                       0                         0                           0
```

Εικόνα 7: One-hot κωδικοποίηση των κατηγορικών χαρακτηριστικών

Οπτικοποίηση Δεδομένων

Ιστογράμματα συχνοτήτων

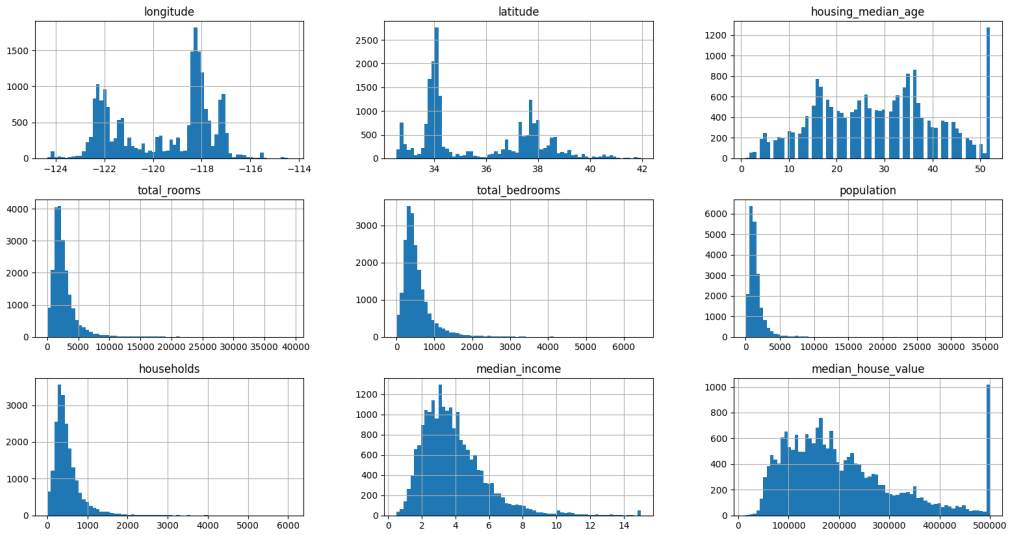
Για την οπτικοποίηση των δεδομένων έχει επιλεχτεί η βιβλιοθήκη `matplotlib.pyplot`.

Για τα αριθμητικά δεδομένα χρησιμοποιείται η συνάρτηση `plot_histogram` με 1 όρισμα.

1. `df (pandas.DataFrame)`: Το dataset που θα χρησιμοποιηθεί για τα ιστογράμματα.

Διαδικασία:

1. Χρήση της συνάρτησης `get_numerical_features` για την επιλογή των αριθμητικών χαρακτηριστικών.
2. Εμφάνιση των ιστογραμμάτων για κάθε αριθμητικό χαρακτηριστικό.



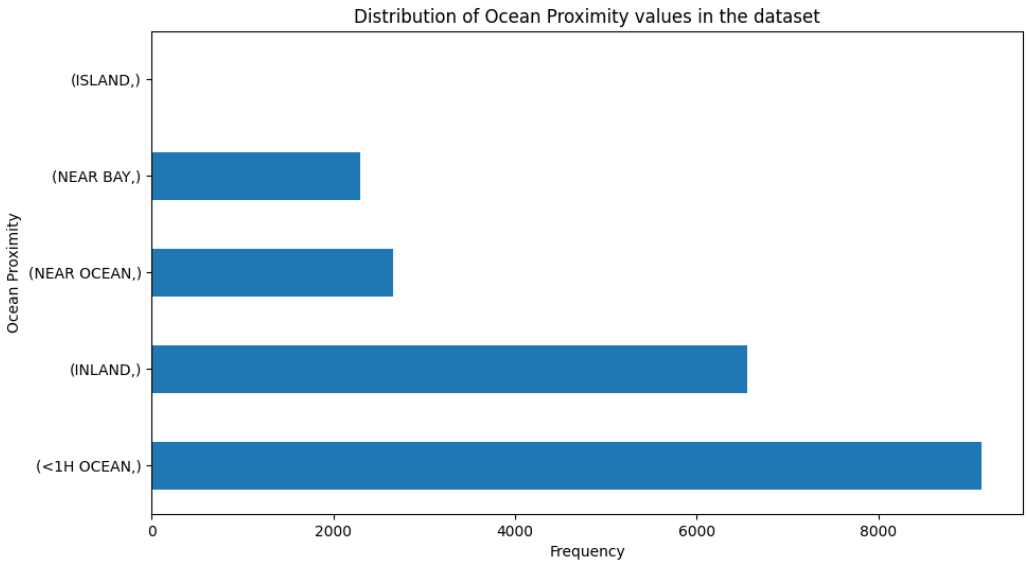
Εικόνα 8: Ιστογράμματα των αριθμητικών χαρακτηριστικών

Για τα κατηγορικά δεδομένα χρησιμοποιείται η συνάρτηση `plot_categorical` με 1 όρισμα.

1. `df (pandas.DataFrame)`: Το dataset που θα χρησιμοποιηθεί για το γράφημα.

Διαδικασία:

1. Χρήση της συνάρτησης `get_categorical_features` για την επιλογή του κατηγορικού χαρακτηριστικού.
2. Υπολογισμός της συχνότητας εμφάνισης των τιμών του κατηγορικού χαρακτηριστικού με την χρήση της μεθόδου `value_counts`.
3. Εμφάνιση γραφήματος ράβδων για τις τιμές του `'ocean_proximity'`.



Εικόνα 9: Γράφημα ράβδων για την κατανομή του κατηγορικού χαρακτηριστικού "ocean_proximity"

Γραφήματα δεδομένων για συνδυασμούς μεταβλητών

Για την οπτικοποίηση των σχέσεων μεταξύ διαφορετικών μεταβλητών επιλέχθηκε η δημιουργία γραφημάτων διασποράς. Για την υλοποίηση των γραφημάτων αξιοποιήθηκαν οι βιβλιοθήκες `seaborn` και `matplotlib.pyplot`.

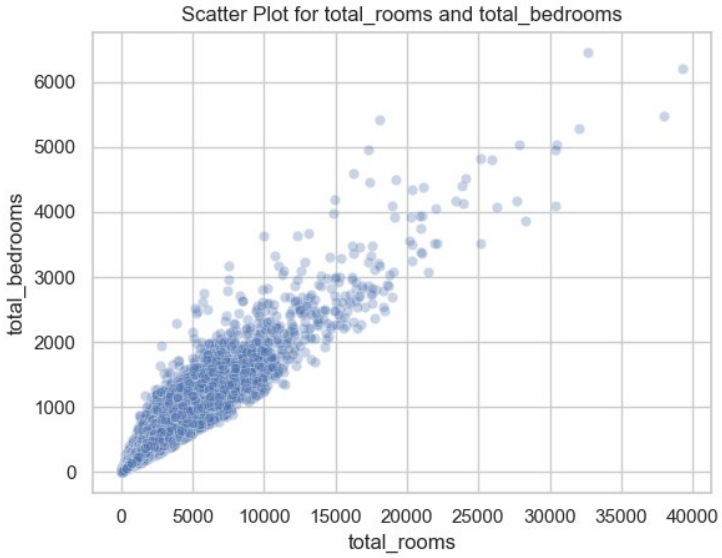
Η συνάρτηση `plot_variable_pairs` δέχεται 2 ορίσματα:

1. `data ( pandas.DataFrame)`: το dataset το οποίο περιέχει τις τιμές των μεταβλητών.
2. `variables (list)`: η λίστα των μεταβλητών οι οποίες θα οπτικοποιηθούν.

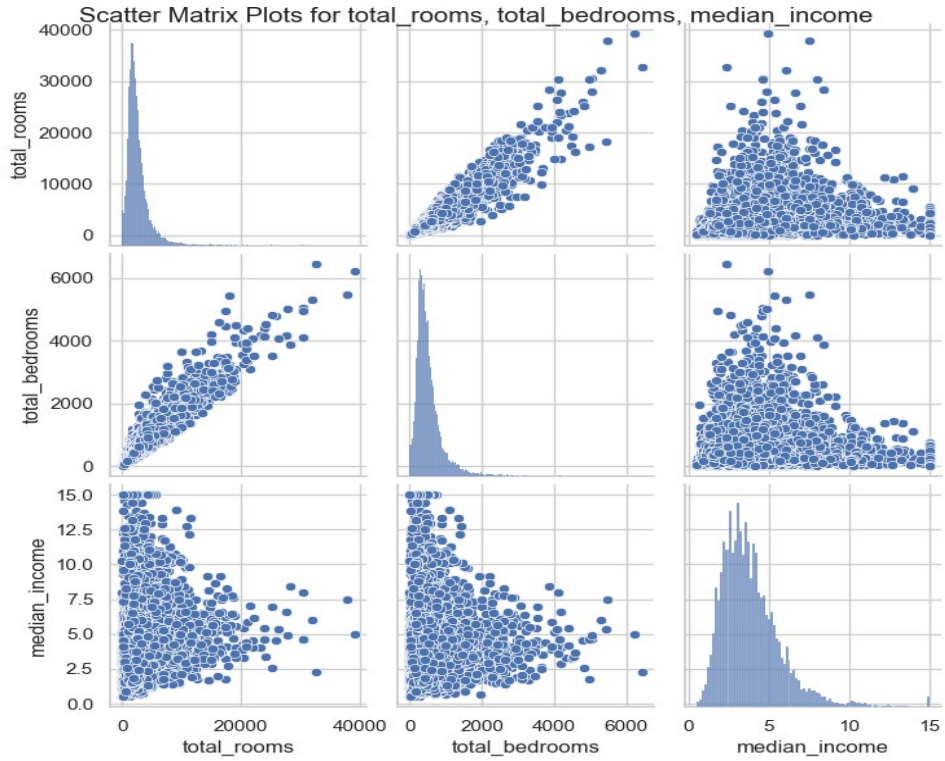
Διαδικασία:

Ελέγχεται αν το πλήθος των μεταβλητών είναι 2:

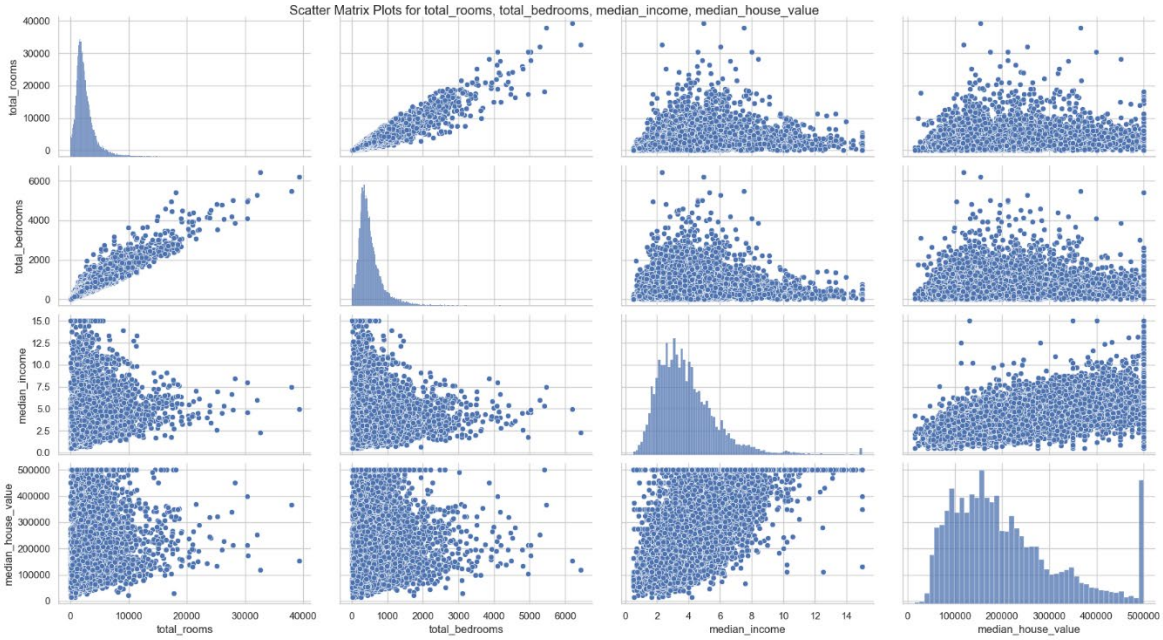
1. Εάν είναι 2 μεταβλητές τότε χρησιμοποιείται η μέθοδος `seaborn.scatterplot`.
2. Αλλιώς αν είναι περισσότερες από 2 μεταβλητές αξιοποιείται η μέθοδος `seaborn.pairplot`.
3. Εμφανίζονται τα αποτελέσματα.



Εικόνα 10: Γράφημα διασποράς για τις μεταβλητές "total_rooms", "total_bedrooms"



Εικόνα 11: Συνδυαστικά γραφήματα για τις 3 μεταβλητές "total_rooms", "total_bedrooms", "median_income"



Εικόνα 12: Συνδυαστικά γραφήματα για τις 4 μεταβλητές "total_rooms", "total_bedrooms", "median_income", "median_house_value"

Παλινδρόμηση δεδομένων

Αλγόριθμος Perceptron

Κύρια συνάρτηση

Η συνάρτηση `perceptron_algorithm` υλοποιεί τον αλγόριθμο μηχανικής μάθησης Perceptron. Δέχεται 3 παραμέτρους:

1. `df (pandas.DataFrame)`: αντιπροσωπεύει τα δεδομένα πάνω στα οποία θα εκπαιδευτεί τον αλγόριθμο.
2. `k (int, Optional)`: αντιπροσωπεύει τον αριθμό των folds για το k-fold Cross-Validation. Εξ' ορισμού έχει τιμή 10.
3. `learning_rate (float, Optional)`: αντιπροσωπεύει τον ρυθμό με τον οποίο ενημερώνονται τα βαρίδια κατά την διάρκεια της εκπαίδευσης του μοντέλου. Εξ' ορισμού έχει τιμή 0.01.

Ο κώδικας της συνάρτησης μπορεί να χωριστεί σε τρία στάδια:

1. Αρχικοποίηση: Εδώ γίνεται η κατάλληλη επεξεργασία των δεδομένων υπολογίζοντας τον μέσο όρο του `target_data` για να χρησιμοποιηθεί σαν κατώφλι (threshold). Επίσης, ξεκινάει η προετοιμασία για το 10-fold Cross-Validation χωρίζοντας τα δεδομένα στα κατάλληλα folds με το κάλεσμα της `split_dataframe_into_folds`. Δημιουργούνται οι λίστες

`target_data_lists` και `input_data_lists` οι οποίες θα περιέχουν τα testing & training set των 10 folds αντίστοιχα. Γίνεται τέλος, αρχικοποίηση των `square_loss_sum` & `absolute_loss_sum` που θα χρησιμοποιηθούν στην συνέχεια για να υπολογιστεί το μέσο τετραγωνικό και μέσο απόλυτο σφάλμα των 10 folds.

2. Με προσπέλαση στην `input_data_lists` και με την χρήση του `fold_index` διατρέχονται όλα τα 10 folds στα οποία έχουν μοιραστεί τα δεδομένα. Αρχικοποιούνται οι τιμές του εκάστοτε μοντέλου το οποίο θα εκπαιδευτεί ορίζοντας το bias ως ένα δεκαδικό αριθμό ανάμεσα στο -100 και στο 100 και τα weights της λίστας weight_data σε δεκαδικούς ανάμεσα στο 0 και στο 1 χρησιμοποιώντας την συνάρτηση `initialize_weight_data`. Στην συνέχεια ορίζονται:
   i. `training_set`: Τα δεδομένα των 9 fold πάνω στα οποία θα εκπαιδευτεί το μοντέλο σε αυτό το iteration του for loop.
   ii. `training_target_set`: Τις επιθυμητές απαντήσεις που αντιστοιχούν στα δεδομένα του training_set. Θα χρησιμοποιηθούν για να υπολογιστεί το σφάλμα.
   iii. `testing_set`: Τα δεδομένα του fold πάνω στα οποία θα δοκιμαστεί το μοντέλο.
   iv. `testing_target_set`: Τις επιθυμητές απαντήσεις που αντιστοιχούν στα δεδομένα του `testing_set`. Θα χρησιμοποιηθούν για να υπολογιστεί το σφάλμα.

   Στην συνέχεια ακολουθεί προσπέλαση στο `training_set`. Αρχικά υπολογίζονται τα `weighted_sum` με κάλεσμα της `calculate_sum`. Έπειτα, καλείται η `activation_function` για να υπολογιστούν οι τελικές προβλέψεις του μοντέλου. Ακολουθεί το βήμα ενημέρωσης των `weight_data` και του bias . Τέλος, γίνεται μια προσπέλαση στο testing_set για να υπολογιστούν με τον ίδιο τρόπο οι προβλέψεις του μοντέλου στα νέα δεδομένα και να υπολογιστεί έτσι το σφάλμα χρησιμοποιώντας τις συναρτήσεις `mean_square_error`, `mean_absolute_error`.

3. Τελική εκπαίδευση του μοντέλου:
   Μετά την ολοκλήρωση του K-fold Cross-Validation επαναλαμβάνεται η διαδικασία της εκπαίδευσης, πρόβλεψης, ενημέρωσης και υπολογισμού σφάλματος σε ολόκληρο το dataset.

Διαχωρισμός του DataFrame σε folds για k-fold Cross-Validation

Ο διαχωρισμός του `DataFrame` σε folds επιτεύχθηκε μέσω της συνάρτησης `split_dataframe_into_folds`. Η συνάρτηση δέχεται 2 παραμέτρους:

1. `df (pandas.DataFrame)`: το `DataFrame` το οποίο θα διαχωριστεί.
2. `k (int, Optional)`: ο αριθμός των fold στα οποία θα διαχωριστεί το `DataFrame`. Εξ' ορισμού έχει τιμή 10.

Διαδικασία:

1. Επιλογή της εξαρτημένης μεταβλητής.
2. Δημιουργία αντιγράφου των δεδομένων.
3. Διαχωρισμός των δεδομένων σε `bins` με βάση την κατανομή των τιμών της εξαρτημένης μεταβλητής, ώστε να υπάρχει ομοιογένεια σε κάθε fold. Δημιουργία μιας νέας στήλης στο `DataFrame` η οποία περιέχει το index του fold στο οποίο ανήκει το κάθε στοιχείο της στήλης της εξαρτημένης μεταβλητής.
4. Shuffling του αντιγράφου του `DataFrame`.

5. Δημιουργία κενού `DataFrame`:
   i. για κάθε fold δημιουργείται κενό `DataFrame current_fold` το οποίο χρησιμοποιείται για να συγκεντρωθούν τα στοιχεία που αντιστοιχούν στο fold.
   ii. υπολογισμός του αριθμού των στοιχείων τα οποία θα επιλεχθούν από το `bin`.
   iii. προσθήκη των στοιχείων στο `current_fold` με την μέθοδο `pandas.concat`.
6. Προσθήκη των τελικών folds στην λίστα `fold_list`.
7. Διαγραφή της στήλης `'bin'` από το κάθε fold.
8. Επιστροφή της τελικής `fold_list`.

Υπολογισμός πρόβλεψης

Για τον υπολογισμό της πρόβλεψης του αλγόριθμου Perceptron απαιτούνται 2 βήματα:

1. Ο υπολογισμός του weighted sum της κάθε γραμμής εισόδου με κάλεσμα της `calculate_sum`. Αυτό επιτυγχάνεται με τον ακόλουθο τύπο:

$$\Sigma_n = X_1W_1 + X_2W_2 + X_3W_3 + \cdots + X_iW_1 + b$$

   Όπου:
   i. Σ_n το weighted sum για n-οστό στοιχείο του πίνακα εισόδου.
   ii. $X_1, X_2, X_3, \dots, X_i$ τα i σε αριθμό στοιχεία εισόδου.
   iii. $W_1, W_2, W_3, \dots, W_i$ τα i σε αριθμό βαρίδια.
   iv. b το bias
2. Λήψη απόφασης με το κάλεσμα της `activation_function`:
   i. 1 εάν το αποτέλεσμα ξεπερνάει το κατώφλι (που έχει υπολογιστεί ως τον μέσο όρο του `median_house_value`)
   ii. -1 εάν το αποτέλεσμα είναι μικρότερο/ίσο

Ενημέρωση των βαρών

Για την ενημέρωση των weight καλείται η `update_weights`. Η `update_weights` δέχεται 5 παραμέτρους:

1. `input_data`: τα δεδομένα που εισάγονται στον αλγόριθμο.
2. `weight_data`: τα βαρίδια που θα ενημερωθούν.
3. `prediction`: η πρόβλεψη του μοντέλου για τα αντίστοιχα δεδομένα εισόδου.
4. `target`: το σωστό αποτέλεσμα για τα αντίστοιχα δεδομένα εισόδου.
5. `learning_rate`: τιμή που αντιπροσωπεύει τον ρυθμό με τον οποίο ενημερώνονται τα weight κατά την διάρκεια της εκπαίδευσης του μοντέλου.

Διαδικασία:

1. Η ενημέρωση των `weight_data` με αριθμό ξεχωριστών βαριδιών n γίνεται με τον ακόλουθο τύπο:

$$W_i' = W_i + a(y - y')X_i$$

   Όπου:
   i. W_i' η νέα τιμή του i -οστού νέου βαριδίου.
   ii. W_i η προηγούμενη τιμή του i-οστου βαριδίου προς ενημέρωση.
   iii. a το `learning_rate`

iv. y το target
v. y' το prediction
vi. X_i i -οστή τιμή εισόδου του αλγόριθμου.

Ενημέρωση του bias

Για την ενημέρωση του bias καλείται η `update_bias`. Η update_bias δέχεται 4 παραμέτρους:

1. `bias`: το bias που θα ενημερωθεί
2. `learning_rate`: αντιπροσωπεύει τον ρυθμό με τον οποίο ενημερώνεται το bias κατά την διάρκεια της εκπαίδευσης του μοντέλου.
3. `prediction`: η πρόβλεψη του μοντέλου για τα αντίστοιχα δεδομένα εισόδου.
4. `target`: το σωστό αποτέλεσμα για τα αντίστοιχα δεδομένα εισόδου.

Διαδικασία:

1. ενημέρωση του `bias` γίνεται με τον παρακάτω τύπο:

$$B' = B + a(y - y')$$

Όπου:

i. B' η νέα τιμή του `bias`
ii. B η παλιά τιμή του `bias`
iii. a το `learning_rate`
iv. y το target
v. y' το prediction

Βοηθητικές συναρτήσεις

Η συνάρτηση `drop_target_data` χρησιμοποιείται για την αφαίρεση των δεδομένων `median_house_value` από τα δεδομένα.

Η συνάρτηση `initialize_target_data` χρησιμοποιείται για την αρχικοποίηση των `target_data` ξεχωρίζοντας τα από τα υπόλοιπα δεδομένα. Στην συνέχεια υπολογίζονται οι κατάλληλες προβλέψεις ελέγχοντας αν κάθε ξεχωριστό `median_house_value` είναι μεγαλύτερο ή μικρότερο ίσου του μέσου όρου. Επιστρέφεται αναλόγως 1 ή -1.

```
Run perceptron:

-------------K-FOLD RESULTS----------------
10-FOLD VALIDATION AVERAGE MSE: 1.8898691226369366
10-FOLD VALIDATION AVERAGE MAE: 0.9449345613184683
------------------------------------------

----------------FINAL MODEL-----------------
FINAL MSE: 0.7467914791490045,FINAL MAE: 0.6770595066509669
```

Εικόνα 13: Αποτελέσματα της εκτέλεσης του αλγορίθμου Perceptron

Αλγόριθμος Ελάχιστου Τετραγωνικού Σφάλματος

Κύρια συνάρτηση

Η συνάρτηση `least_squares_algorithm` υλοποιεί τον αλγόριθμο παλινδρόμησης Least Squares. Δέχεται 2 παραμέτρους:

1. `input_data (pandas.DataFrame)`: αντιπροσωπεύει τα δεδομένα πάνω στα οποία θα εκπαιδευτεί ο αλγόριθμος.
2. `num_folds (int)`: αντιπροσωπεύει τον αριθμό των folds για το k-fold Cross-Validation.

Αρχικοποίηση:

1. Εδώ γίνεται η κατάλληλη επεξεργασία των δεδομένων αρχικοποιώντας το `skf` με το κάλεσμα της `StratifiedKFold` αλλά και τους πίνακες `X` και `y_binned` όπου αντιπροσωπεύουν:
   i. `X`: Δεδομένα πάνω στα οποία θα εκπαιδευτεί ο αλγόριθμος.
   ii. `y_binned`: Την εξαρτημένη μεταβλητή με βάση την κατανομή των τιμών της. Θα χρησιμοποιηθεί μαζί με τις προβλέψεις του μοντέλου για να υπολογιστεί το σφάλμα.
2. Στην συνέχεια αρχικοποιούνται οι ακόλουθες μεταβλητές:
   i. `training_mae_scores`: Το άθροισμα του μέσου απόλυτου σφάλματος σε όλα τα training folds.
   ii. `training_mse_scores`: Το άθροισμα του μέσου τετραγωνικού σφάλματος σε όλα τα training folds.
   iii. `testing_mae_scores`: Το άθροισμα του μέσου απόλυτου σφάλματος σε όλα τα testing folds.
   iv. `testing_mse_scores`: Το άθροισμα του μέσου τετραγωνικού σφάλματος σε όλα τα testing folds.

K-fold Cross-Validation:

1. Υπολογίζονται για αρχή τα αντίστοιχα training και testing sets για αυτό το iteration του k-fold Cross-Validation ως:
   i. `X_train`: Τα δεδομένα που θα χρησιμοποιηθούν για να εκπαιδευτεί ο αλγόριθμος.
   ii. `X_test`: Τα δεδομένα που θα χρησιμοποιηθούν κατά τον έλεγχο.
   iii. `y_train`: Τα επιθυμητά αποτελέσματα του αντίστοιχου `X_train`. Με αυτά και τις προβλέψεις θα υπολογιστεί το σφάλμα.
   iv. `y_test`: Τα επιθυμητά αποτελέσματα του αντίστοιχου `X_test`. Με αυτά και τις προβλέψεις θα υπολογιστεί το σφάλμα.
2. Υπολογίζεται ο πίνακας κλίσης ευθείας B.
3. Υπολογίζεται ο πίνακας προβλέψεων Y'.
4. Στην συνέχεια υπολογίζονται με την βοήθεια των `calculate_square_error_matrix` & `calculate_absolute_error_matrix` το τετραγωνικό και απόλυτο σφάλμα και λαμβάνεται ο μέσος όρος. Ο υπολογισμός των προβλέψεων επαναλαμβάνεται με τα

δεδομένα τα οποία περιέχονται στις μεταβλητές `X_test_array` και `Y_test_array`, χρησιμοποιώντας τον πίνακα B που υπολογίστηκε από τα δεδομένα του σετ εκπαίδευσης, για να εντοπιστούν τα σφάλματα κατά τη διάρκεια του testing. Η διαδικασία ολοκληρώνεται για κάθε fold μεμονωμένα.

Τέλος, με την ολοκλήρωση του k-fold Cross-Validation, γίνεται υπολογισμός του τελικού πίνακα B χρησιμοποιώντας ολόκληρο το dataset.

Υπολογισμός line of best fit

Για τον υπολογισμό του line of best fit και με δεδομένο ότι οι ευθείες οι οποίες υπολογίζονται ξεκινούν από την αρχή των αξόνων $(0,0)$ θα έχουν τύπο ευθείας:

$$y' = X_1 b_1 + X_2 b_2 + \cdots + X_i b_i$$

Όπου:

1. $X_1, X_2, X_{3,\dots,} X_i$ οι τιμές των εισόδων του αλγορίθμου οι οποίες αποτελούν και τις τιμές των i διαφορετικών αξόνων στο υπερεπίπεδο.
2. $b_1, b_2, b_{3,\dots,} b_i$ οι τιμές κλίσης της ευθείας για κάθε έναν από τους i άξονες του υπερεπίπεδου.
3. y' η τιμή που πρόβλεψε το μοντέλο με εισόδους $X_1, X_2, X_{3,\dots,} X_i$.

Με τη χρήση πινάκων ο παραπάνω τύπος τροποποιείται ως εξής:

$$Y' = XB^T$$

Όπου:

1. Y' ο πίνακας ο οποίος περιέχει τις προβλέψεις με συνολικό αριθμό n ίσο με τα διαφορετικά δείγματα των δεδομένων.
2. X ο πίνακας ο οποίος περιέχει στήλες i ίσες σε αριθμό με τον αριθμό των χαρακτηριστικών τα οποία δίνονται στο μοντέλο και γραμμές με αριθμό n ίσο με τα διαφορετικά δείγματα που των δεδομένων.
3. B^T ο ανάστροφος του πίνακα κλίσης ευθείας B με i στήλες με το i ίσο με τον αριθμό των χαρακτηριστικών.

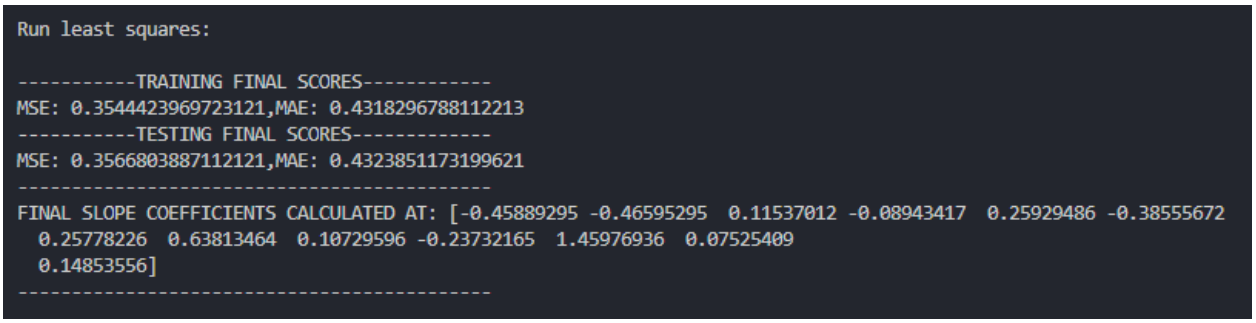
Υπολογισμός πίνακα κλίσης ευθείας B

Ο πίνακας κλίσης ευθείας B υπολογίζεται με τον ακόλουθο τύπο μέσω της συνάρτησης `calculate_slope_coefficient_matrix`:

$$B = (X^T X)^{-1}(X^T Y)$$

Όπου:

1. X^T ο ανάστροφος πίνακας του X.
2. X ο πίνακας των δεδομένων ο οποίος παρέχεται σαν είσοδος στον αλγόριθμο.
3. Y ο πίνακας των επιθυμητών αποτελεσμάτων/στόχων.



Εικόνα 14: Αποτελέσματα της εκτέλεσης του αλγορίθμου Least Squares

Υπολογισμός σφάλματος

Υπολογισμός Mean Square Error

Ο υπολογισμός του μέσου τετραγωνικού σφάλματος επιτυγχάνεται με το κάλεσμα των συναρτήσεων `mean_square_error` στην υλοποίηση του Perceptron και `calculate_square_error_matrix` στην υλοποίηση του Least Squares. Η πρώτη συνάρτηση υπολογίζει το σφάλμα για κάθε i-οστό στοιχείο ξεχωριστά ενώ η δεύτερη κάνει πράξεις πινάκων με την χρήση `NumPy` arrays.

Ο γενικός τύπος υπολογισμού του squared error είναι:

$$E_i = (Y_i - Y_i')^2$$

Όπου:

1. E_i το σφάλμα για την i-οστή πρόβλεψη του μοντέλου.
2. Y_i το σωστό αποτέλεσμα για το i-οστό στοιχείο του πίνακα εισόδου.
3. Y_i' η πρόβλεψη που έδωσε το μοντέλο για το i-οστό στοιχείο του πίνακα εισόδου.

Στην συνέχεια το άθροισμα των σφαλμάτων διαιρείται με το πλήθος των δειγμάτων n. Ο τελικός τύπος διαμορφώνεται ως εξής:

$$MSE = \frac{1}{n}\sum_{i=1}^{n} E_i$$

Υπολογισμός Mean Absolute Error

Ο υπολογισμός του μέσου απολύτου σφάλματος επιτυγχάνεται με το κάλεσμα των συναρτήσεων `mean_absolute_error` στην υλοποίηση του Perceptron και `calculate_absolute_error_matrix` στην υλοποίηση του Least Squares. Η πρώτη συνάρτηση υπολογίζει το σφάλμα για κάθε i-οστό στοιχείο ξεχωριστά ενώ η δεύτερη κάνει πράξεις πινάκων με την χρήση `NumPy` arrays.

Ο γενικός τύπος υπολογισμού του absolute error είναι:

$$E_i = |Y_i - Y_i'|$$

Ο τελικός τύπος διαμορφώνεται ως εξής:

$$MAE = \frac{1}{n}\sum_{i=1}^{n} E_i$$

Πολυστρωματικό νευρωνικό δίκτυο

Η συνάρτηση `mlp_regression` υλοποιεί την παλινδρόμηση Multilayer Perceptron στο dataset. Χρησιμοποιεί την μέθοδο k-fold Cross-Validation επαλήθευση. Δέχεται 2 ορίσματα:

1. `df (pandas.DataFrame)`: Το σύνολο δεδομένων το οποίο θα χρησιμοποιηθεί για την εκπαίδευση και την επικύρωση του μοντέλου.
2. `num_folds (int, Optional)`: Ο αριθμός των fold για την k-fold επικύρωση. Εξ' ορισμού έχει τιμή 10.

Διαδικασία:

1. Τα δεδομένα χωρίζονται σε features (`X`) και target (`y`)
2. Χρήση της μεθόδου `pandas.qcut` για να επιτευχθεί ομοιόμορφη κατανομή των στοιχείων της εξαρτημένης μεταβλητής y.
3. Αρχικοποίηση του `MLPRegressor` με 100 neurons, `random state` 42 (για την αρχικοποίηση των βαρών). Τερματίζει μετά από 1000 επαναλήψεις ή εφόσον επέλθει σύγκλιση.
4. Αρχικοποίηση διαδικασίας της διασταύρωσης με την χρήση του `StratifiedKFold`, για τη διασφάλιση ομοιογενούς αναπαράστασης των τιμών της εξαρτημένης μεταβλητής `y` σε κάθε fold.
5. Εκπαίδευση και Επικύρωση: Το μοντέλο εκπαιδεύεται σε κάθε fold και υπολογίζονται οι μετρήσεις `MSE` και `MAE` τόσο στην φάση της εκπαίδευσης όσο και στην φάση της επικύρωσης.
6. Μέσοι Όροι Μετρήσεων: Υπολογισμός και εμφάνιση των μέσων τιμών `MSE` και `MAE` για την εκπαίδευση και την επικύρωση σε όλα τα folds.
7. Εκπαίδευση στο πλήρες σύνολο δεδομένων: Μετά την ολοκλήρωση της k-fold Cross-Validation, το μοντέλο εκπαιδεύεται εκ νέου στο πλήρες σύνολο δεδομένων για την μέγιστη αξιοποίηση των διαθέσιμων πληροφοριών.
8. Αποτελέσματα στο πλήρες σύνολο δεδομένων: Υπολογισμός και εμφάνιση των `MSE` και `MAE` μετά την εκπαίδευση στο πλήρες σύνολο δεδομένων, προσδίδοντας μια ενδεικτική αξιολόγηση της πραγματικής απόδοσης του μοντέλου.
9. Επιστροφή του τελικού μοντέλου.

```
Run mlp regression:

Average Training MSE: 0.1909673194641204
Average Training MAE: 0.29767034973110473
Average Validation MSE: 0.21272828862918533
Average Validation MAE: 0.31164570330421576
MSE after training on the entire dataset: 0.18817089308118115
MAE after training on the entire dataset: 0.29916321554584246
```

Εικόνα 15: Αποτελέσματα της εκτέλεσης του Multi-Layer-Perceptron

Ευρετήριο Εικόνων

Βιβλιογραφία

Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., … Oliphant, T. E. (2020). Array programming with NumPy. Nature, 585(7825), 357–362. https://doi.org/10.1038/s41586-020-2649-2

Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. Computing in Science & Engineering, 9(3), 90–95. https://doi.org/10.1109/mcse.2007.55

Matplotlib development team. (n.d.). matplotlib.pyplot — Matplotlib 3.5.3 documentation. Matplotlib — Visualization with Python. https://matplotlib.org/3.5.3/api/_as_gen/matplotlib.pyplot.html

NumPy. (n.d.). numpy.linalg.inv — NumPy v1.26 Manual. NumPy -. https://numpy.org/doc/stable/reference/generated/numpy.linalg.inv.html

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., … Duchesnay, E. (2011). Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research, 12, 2825–2830.

Scikit Learn. (n.d.-a). sklearn.neural_network.MLPRegressor. scikit-learn. https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPRegressor.html

Scikit Learn. (n.d.-b). sklearn.model_selection.StratifiedKFold. scikit-learn. https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.StratifiedKFold.html

Scikit Learn. (n.d.-c). sklearn.preprocessing.MinMaxScaler. scikit-learn. https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html

Scikit Learn. (n.d.-d). sklearn.preprocessing.OneHotEncoder. scikit-learn. https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.OneHotEncoder.html

Scikit Learn. (n.d.-e). sklearn.preprocessing.RobustScaler. scikit-learn. https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.RobustScaler.html

Scikit Learn. (n.d.-f). sklearn.preprocessing.StandardScaler. scikit-learn. https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html#sklearn.preprocessing.StandardScaler

Seaborn development team. (n.d.-a). seaborn.pairplot — seaborn 0.13.2 documentation. seaborn: statistical data visualization — seaborn 0.13.2 documentation. https://seaborn.pydata.org/generated/seaborn.pairplot.html

Seaborn development team. (n.d.-b). seaborn.scatterplot — seaborn 0.13.2 documentation. seaborn: statistical data visualization — seaborn 0.13.2 documentation. https://seaborn.pydata.org/generated/seaborn.scatterplot.html

The pandas development team. (2024). pandas-dev/pandas: Pandas (v2.2.0). Zenodo. https://doi.org/10.5281/zenodo.10537285

The pandas development team. (n.d.-a). pandas.concat — pandas 2.2.1 documentation. pandas - Python Data Analysis Library. https://pandas.pydata.org/docs/reference/api/pandas.concat.html

The pandas development team. (n.d.-b). pandas.DataFrame.select_dtypes — pandas 2.2.1 documentation. pandas - Python Data Analysis Library. https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.select_dtypes.html

The pandas development team. (n.d.-c). pandas.qcut — pandas 2.2.1 documentation. pandas - Python Data Analysis Library. https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.qcut.html

The pandas development team. (n.d.-d). pandas.read_csv — pandas 2.2.1 documentation. pandas - Python Data Analysis Library. https://pandas.pydata.org/docs/reference/api/pandas.read_csv.html

The pandas development team. (n.d.-e). pandas.Series.value_counts — pandas 2.2.1 documentation. pandas - Python Data Analysis Library. https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.Series.value_counts.html

Waskom, M. L. (2021). seaborn: statistical data visualization. Journal of Open Source Software, 6(60), 3021. https://doi.org/10.21105/joss.03021