ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ
Τμήμα Πληροφορικής

Εργασία Μαθήματος «Αναγνώριση Προτύπων»

Απαλλακτική Εργασία Μαθήματος	Στόχος της συγκεκριμένης εργασίας είναι η ανάπτυξη αλγορίθμων μηχανικής μάθησης για την προσέγγιση της διάμεσης τιμής ενός ακινήτου σε μια ευρύτερη γεωγραφική περιοχή της πολιτείας της Καλιφόρνια βάσει ενός συνόλου αντικειμενικών χαρακτηριστικών των ακινήτων στην εν λόγω περιοχή. Κάθε τέτοια περιοχή αποτελεί στην πραγματικότητα την μικρότερη γεωγραφική οντότητα που καταγράφηκε στην σχετική απογραφή του 1990 με πληθυσμιακό εύρος μεταξύ των 600 και 3000 κατοίκων.
Εκπαιδευτές	**Δ. Σωτηρόπουλος , Γ.Τσιχριντζής**
Όνομα φοιτητή – Αρ. Μητρώου	Αργυροπούλου Μαρία : Π18011
	Γκινετσι Αλέξανδρος: Π18028
	Εκατομμάτης Αντώνης : Π18039
Ημερομηνία παράδοσης	22-02-2022

Περιεχόμενα

Εισαγωγή

Για την ανάπτυξή των προγραμμάτων κάναμε την επιλογή να χρησιμοποιήσουμε το λογισμικό PyCharm και για γλώσσα προγραμματισμού τη Python (3.9), λόγο της εξοικείωσής μας με την γλώσσα. Έγινε επίσης χρήση των βιβλιοθηκών Numpy, pandas, matplotlib, mlxtend και sklearn.

Προ-επεξεργασία Δεδομένων

Αρχικά διακρίναμε τα υποσύνολα των αριθμητικών και των κατηγορικών χαρακτηριστικών που μας δόθηκαν για επεξεργασία.

Αριθμητικά χαρακτηριστικά	**Κατηγορικά χαρακτηριστικά.**
Longitude	Ocean_proximity
Latitude	
Housing_median_age	
Total_rooms	
Total_bedrooms	
Population	
Households	
Median_income	

Η διάμεση τιμή των ακινήτων της περιοχής (median_house_value) που θέλουμε να προσεγγίσουμε είναι επίσης αριθμητικό χαρακτηριστικό.
Η τεχνική κλιμάκωσης (scaling) που επιλέξαμε να χρησιμοποιήσουμε είναι η min-max scalling της βιβλιοθήκης mlxtend. Τα δεδομένα κλιμακώνονται σε ένα σταθερό εύρος - συνήθως 0 έως 1. Ο λόγος ύπαρξης αυτού του οριοθετημένου εύρους - σε αντίθεση με το standardization - είναι ότι θα καταλήξουμε σε μικρότερες τυπικές αποκλίσεις, οι οποίες μπορούν να καταστείλουν την επίδραση των ακραίων τιμών.

Μια κλιμάκωση Min-Max γίνεται συνήθως μέσω της ακόλουθης εξίσωσης:

$$X_{sc} = \frac{X - X_{min}}{X_{max} - X_{min}}.$$

Αυτή η βιβλιοθήκη, επιπλέον, σε περίπτωση ελλιπών τιμών δίνει αποτέλεσμα Nan, με αποτέλεσμα να μπορούμε εύκολα να αναγνωρίσαμε τις ελλιπείς τιμές και συγκεκριμένα στα δικά μας δεδομένα, τις ελλείψεις στο χαρακτηριστικό total_bedrooms. Αυτές τις συγκεκριμένες εγγραφές τις συμπληρώσαμε, με την συνάρτηση fillna, με την διάμεση τιμή του χαρακτηριστικού.
Για να αναπαραστήσουμε το κατηγορικό χαρακτηριστικό χρησιμοποιούμε την αναπαράσταση One Hot Vector ώστε να λάβουν τα δεδομένα διανυσματική

αναπαράσταση. Μια One Hot Vector κωδικοποίηση είναι μια αναπαράσταση κατηγορικών μεταβλητών ως δυαδικά διανύσματα. Αυτό απαιτεί πρώτα να αντιστοιχιστούν οι κατηγορικές τιμές σε ακέραιες τιμές (χρήση της LabelEncoder()). Στη συνέχεια, κάθε ακέραια τιμή αντιπροσωπεύεται ως ένα δυαδικό διάνυσμα που είναι όλες μηδενικές τιμές εκτός από τον δείκτη του ακέραιου, ο οποίος σημειώνεται με 1.

Επομένως κάθε εγγραφή του χαρακτηριστικού ocean proximity θα κωδικοποιηθεί στην μορφή:

- [1.0.0.0.0] = <1H Ocean
- [0.1.0.0.0] = InLand
- [0.0.1.0.0] = IsLand
- [0.0.0.1.0] = Near Bay
- [0.0.0.0.1] = Near ocean

Ολοκληρώνοντας τον μετασχηματισμό των δεδομένων είναι απαραίτητο πριν χρησιμοποιήσουμε αλγόριθμους ταξινόμησης (classifiers) των δεδομένων να διαχωρίσουμε τα δεδομένα σε σύνολα εκπαίδευσης και δοκιμών.

Έτσι για κάθε μηχανισμό μηχανικής μάθησης θα χρησιμοποιήσουμε την διαδικασία κατάτμησης των δεδομένων, σε υποσύνολα εκπαίδευσης και ελέγχου, της μεθόδου της 10-πλής διεπικύρωσης (10 fold cross validation).

Δηλαδή το σύνολο των δεδομένων μας θα διαμοιραστεί σε 10 υποσύνολα. εκ των οποίων καθένα από τα υποσύνολα θα εμπεριέχει 90% για training και 10% των δεδομένων θα χρησιμοποιείται για testing.

Στο πρώτο fold το πρώτο 10% των δεδομένων θα χρησιμοποιηθεί για έλεγχο (test) και το υπόλοιπο 90% για εκπαίδευση (train). Στο
2 fold το 2^{ο} 10% των δεδομένων θα χρησιμοποιηθεί για έλεγχο (test) και το υπόλοιπο 90% για εκπαίδευση (train). Στο 3 fold το 3^{ο} 10% των δεδομένων θα χρησιμοποιηθεί για έλεγχο (test) και το υπόλοιπο 90% για εκπαίδευση (train). Μέχρι που στο τέλος στο 10^{ο} fold το πρώτο 90% των δεδομένων θα χρησιμοποιηθεί για εκπαίδευση (train) και το υπόλοιπο 10 % για έλεγχο (test).

Οπτικοποίηση Δεδομένων

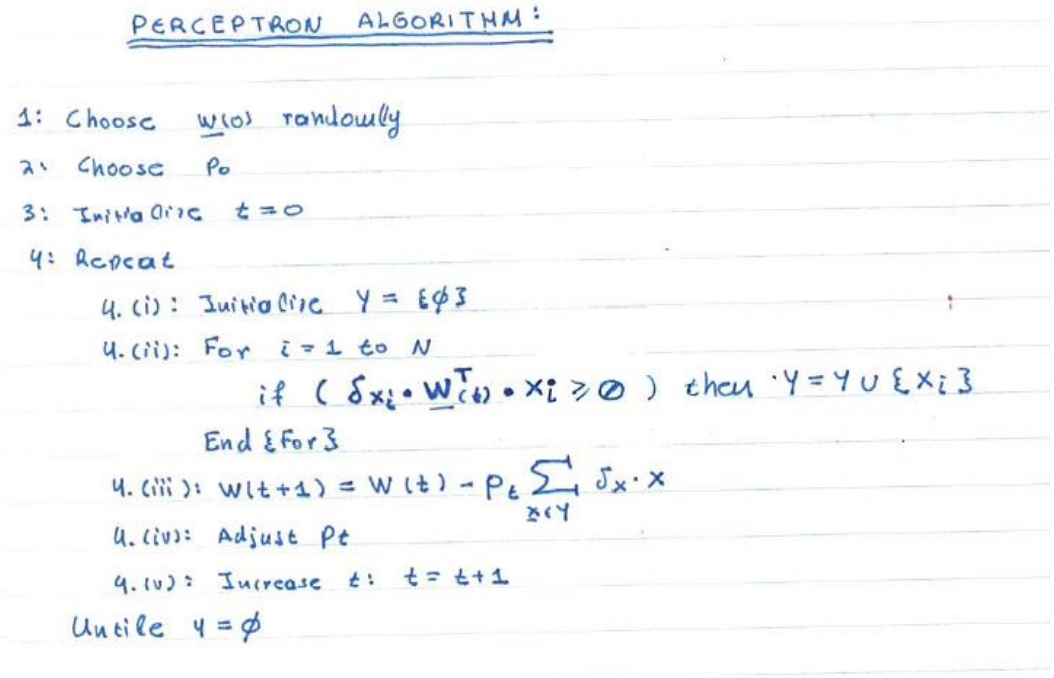
Για να αναπαραστήσουμε γραφικά τα ιστογράμματα συχνοτήτων (που αντιστοιχούν στις συναρτήσεις πυκνότητας πιθανότητας) για κάθε μία από τις 10 μεταβλητές του dataframe df χρησιμοποιούμαι την βιβλιοθήκη matplotlib και seaborn και πιο συγκεκριμένα τις μεθόδους pyplot, histplot και countplot . Δημιουργούμε ένα subplot με 9 γραφήματα στην κάθετη διάσταση και 2 στην οριζόντια διάσταση, μεγέθους 10x8. Στη συνέχεια, για κάθε μια από τις 10 μεταβλητές, δημιουργούμε ένα ιστόγραμμα στο αντίστοιχο subplot. Το ιστόγραμμα δεξιά από το κάθε αρχικό

ιστόγραμμα αντιπροσωπεύει το ιστόγραμμα της ίδιας μεταβλητής αλλά μετά από κλιμάκωση (scaling) της τιμής της μεταβλητής. Το κατηγορικό χαρακτηριστικό το εμφανίζουμε σε ξεχωριστό παράθυρο για την καλύτερη διάκριση του.

Για να αναπαραστήσουμε δισδιάστατα γραφήματα δεδομένων), χρησιμοποιούμε τη μέθοδο scatterplot από τη βιβλιοθήκη matplotlib.
Ξεκινάμε ενώνοντας τα κλιμακούμενα(scaled) αριθμητικά χαρακτηριστικά και το κωδικοποιημένο κατηγορικό χαρακτηριστικό σε έναν ενιαίο πίνακα NumPy, μετατρέπουμε τον πίνακα αυτό σε Pandas DataFrame και εκχωρούμε τα ονόματα στηλών σε κάθε χαρακτηριστικό.
Το πρώτο γράφημα είναι ένα δισδιάστατο διάγραμμα διασποράς (scatterplot), με 2 μεταβλητές, γεωγραφικού μήκους (longitude) και γεωγραφικού πλάτους (latitude). Με κάθε σημείο χρωματισμένο κόκκινο και το χρώμα των άκρων σε μαύρο. Το δεύτερο γράφημα είναι μια διασπορά του πληθυσμού (population) και του μέσου εισοδήματος (median_income). Το τρίτο γράφημα είναι ένα δισδιάστατο διάγραμμα διασποράς με 3 μεταβλητές. Του συνολικού πλήθους δωματίων (total_rooms), με το συνολικό πλήθος υπνοδωματίων (total_bedrooms) και με το μέγεθος κάθε σημείου να καθορίζεται από τον αριθμό των νοικοκυριών (households). Τέλος το τέταρτο γράφημα, δισδιάστατο γράφημα που αναπαριστά 4 μεταβλητές. Είναι μια διασπορά των νοικοκυριών (households)και του μέσου εισοδήματος (median_income), με το μέγεθος κάθε σημείου να καθορίζεται από τη διάμεση ηλικία κατοικίας (housing_median_age) και το χρώμα του να ορίζεται από την κατηγορία εγγύτητας ωκεανού (ocean_proximity). Δεξιά από το διάγραμμα προστίθεται μια γραμμή χρώματος (colorbar) για να εμφανιστεί το εύρος τιμών για την κατηγορία εγγύτητας ωκεανού.

Υλοποίηση αλγόριθμου Perceptron

Η υλοποίηση του βρίσκεται στο ξεχωριστό αρχείο **perceptron.py** και ενσωματώνεται στο κεντρικό πρόγραμμα **main.py**. Η ανάπτυξη του Perceptron βασίστηκε στον παρακάτω αλγόριθμο.



Ο κώδικας ορίζει μια κλάση Perceptron με τις μεθόδους fit() και predict() για δυαδική ταξινόμηση. Στον constructor της κλάσης ορίζονται οι προαιρετικές παράμετροί του: learning_rate = 0.1 και num_iterations = 50 και αρχικοποιούνται τα βάρη (weights) της g. Στη μέθοδο fit(), αρχικά δίνονται τυχαίες τιμές στα βάρη χρησιμοποιώντας τη μέθοδο np.random.rand() και ο αλγόριθμος επαναλαμβάνεται για τον αριθμό επαναλήψεων που έχει οριστεί προηγουμένως. Σε κάθε επανάληψη, ο αλγόριθμος υπολογίζει το dot product των βαρών και των δεδομένων εισόδου. Αν το γινόμενο έχει διαφορετικό πρόσημο από το στόχο y, μετασχηματίζονται τα βάρη με σκοπό να αλλάξει η κλήση της ευθείας g. Το learning_rate μικραίνει καθώς πολλαπλασιάζεται με έναν παράγοντα 0,95 σε κάθε επανάληψη και αυξάνεται και ο μετρητής t. Στη μέθοδο predict(), ο αλγόριθμος επιστρέφει τις προβλεπόμενες τιμές του y για τα δεδομένα εισόδου χρησιμοποιώντας τα βάρη που υπολογίστηκαν κατά την διάρκεια του training.

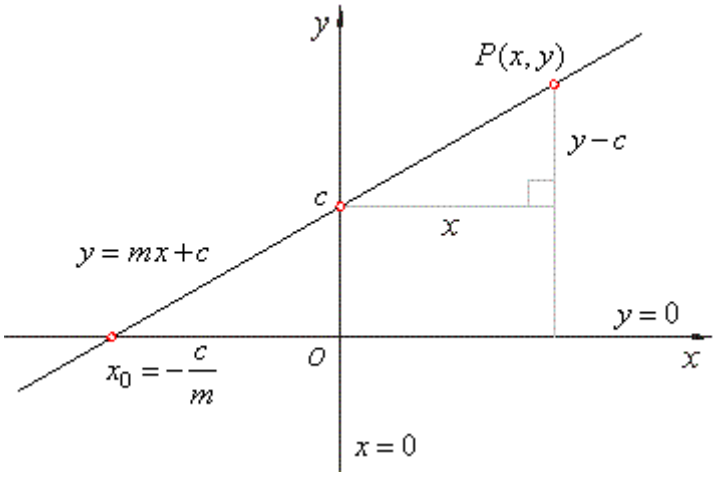
Αλγόριθμος Ελαχίστου Τετραγωνικού Σφάλματος (Least Squares)

Η υλοποίηση του βρίσκεται στο ξεχωριστό αρχείο **leastsquares.py** και ενσωματώνεται στο κεντρικό πρόγραμμα **main.py**

Η Γραμμική παλινδρόμηση είναι μια από τις απλούστερες μορφές μηχανικής μάθησης. Εδώ, θα εξετάσουμε πώς λειτουργεί η γραμμική παλινδρόμηση και θα την εφαρμόσουμε χρησιμοποιώντας των γλώσσα προγραμματισμού Python.
Στη στατιστική, η γραμμική παλινδρόμηση είναι μια γραμμική προσέγγιση για τη μοντελοποίηση της σχέσης μεταξύ μιας εξαρτημένης μεταβλητής και μιας ή

περισσότερων ανεξάρτητων μεταβλητών. Στην περίπτωση μιας ανεξάρτητης μεταβλητής ονομάζεται απλή γραμμική παλινδρόμηση. Για περισσότερες από μία ανεξάρτητες μεταβλητές, η διαδικασία ονομάζεται πολλαπλή γραμμική παλινδρόμηση. Εδώ θα εξετάσουμε μια περίπτωση απλής γραμμικής παλινδρόμησης Έστω Χ η ανεξάρτητη μεταβλητή και Υ η εξαρτημένη μεταβλητή. Θα ορίσουμε μια γραμμική σχέση μεταξύ αυτών των δύο μεταβλητών ως εξής:

$$Y = mX + c$$



Παραπάνω μια απλή εξίσωση γραμμής. m είναι η κλίση της ευθείας και c είναι η τομή y. Θα χρησιμοποιήσουμε αυτήν την εξίσωση για να εκπαιδεύσουμε το πρόγραμμα μας με ένα ήδη υπάρχων σύνολο δεδομένων και να προβλέψουμε την τιμή του Υ για οποιαδήποτε δεδομένη τιμή του Χ. Θα να προσδιορίσουμε την τιμή των m και c, που δίνει το ελάχιστο σφάλμα για το σύνολο δεδομένων. Θα το κάνουμε αυτό χρησιμοποιώντας τη μέθοδο του ελαχίστου τετραγώνου(Least Squares).

Εύρεση του Σφάλματος

Για να ελαχιστοποιήσουμε το σφάλμα χρειαζόμαστε έναν τρόπο να υπολογίσουμε το σφάλμα εξαρχής. Μια συνάρτηση απώλειας στη μηχανική μάθηση είναι απλώς ένα μέτρο του πόσο διαφορετική είναι η προβλεπόμενη τιμή από την πραγματική τιμή.

Σε αυτό το σημείο, θα χρησιμοποιήσουμε τη Συνάρτηση Τετραγωνικής Απώλειας για να υπολογίσουμε την απώλεια ή το σφάλμα στο πρόγραμμα μας. Μπορεί να οριστεί ως:

$$L(x) = \sum_{i=1}^{n} (y_i - p_i)^2$$

Μέθοδος των ελαχίστων τετραγώνων (Least Squares)
Τώρα που προσδιορίσαμε τη συνάρτηση απώλειας, το μόνο που μένει να κάνουμε είναι να την ελαχιστοποιήσουμε. Αυτό γίνεται βρίσκοντας τη μερική παράγωγο του L, εξισώνοντάς την με 0 και στη συνέχεια βρίσκοντας μια έκφραση για τα m και c. Αφού κάνουμε τα μαθηματικά, μένουμε με αυτές τις εξισώσεις:

$$m = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

$$c = \bar{y} - m\bar{x}$$

Εδώ ως $\bar{x}$ ορίζεται ως ο μέσος όρος όλων των τιμών στην είσοδο X και $\bar{y}$ ως ο μέσος όρος όλων των τιμών στην επιθυμητή έξοδο Y. Αυτή είναι η μέθοδος των ελάχιστων τετραγώνων.

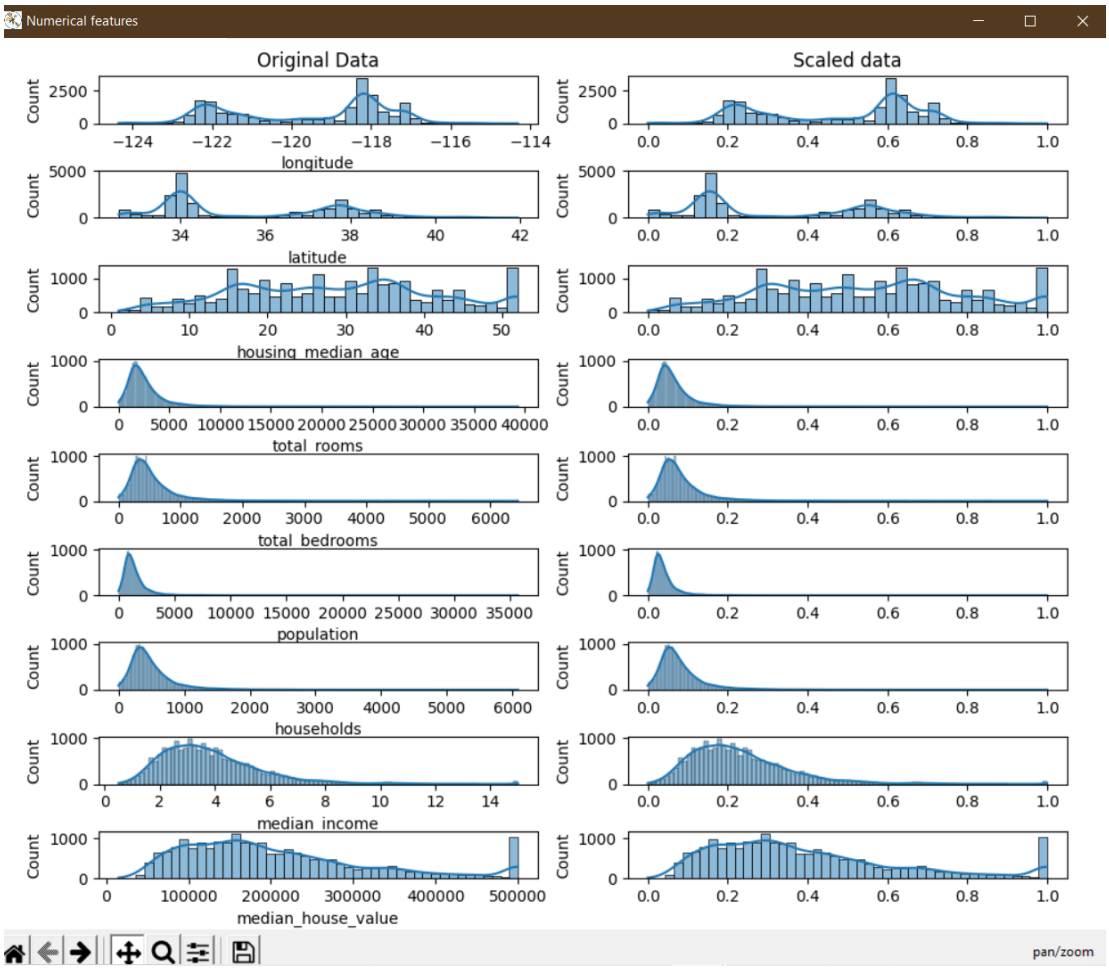
Πολυστρωματικό νευρωνικό δίκτυο

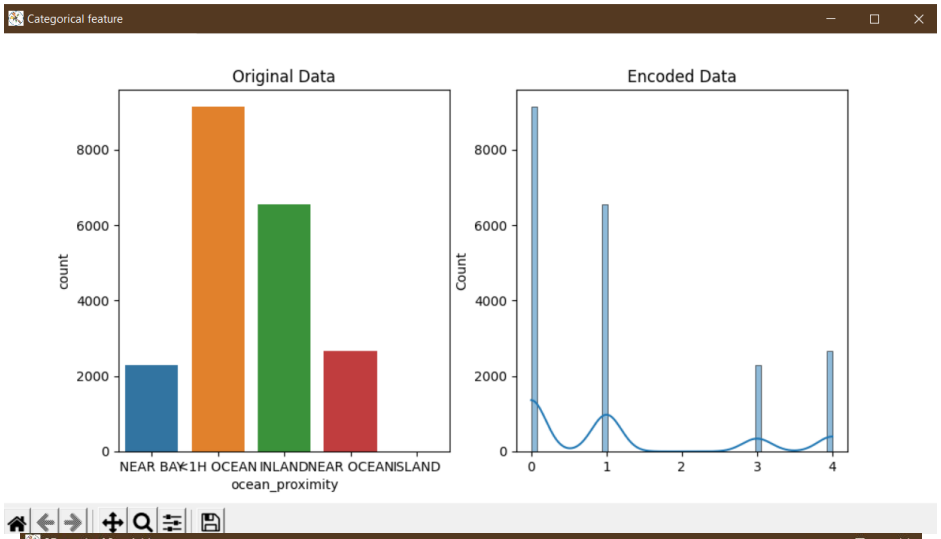
Το mlp.py

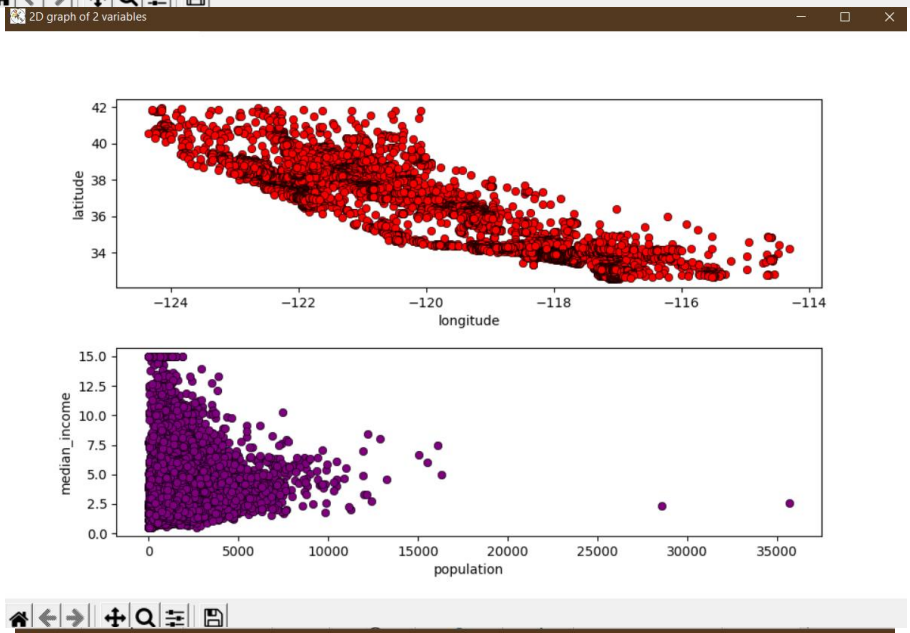
Εισάγουμε τις απαραίτητες βιβλιοθήκες με λειτουργίες και κλάσεις, όπως τον ColumnTransformer, το SimpleImputer, το StandardScaler, το OneHotEncoder και το MLPRregressor. Φορτώνουμε τα δεδομένα από το αρχείο house.csv σε ένα πλαίσιο δεδομένων pandas χρησιμοποιώντας το read_csv λειτουργία. Χωρίζουμε τα δεδομένα σε χαρακτηριστικά (features) και στόχο (target). Τα features είναι όλες οι στήλες εκτός από τη στήλη median_house_value, η οποία είναι ο στόχος που θέλουμε να προβλέψουμε. Χωρίζουμε τα δεδομένα σε σύνολα εκπαίδευσης (training set) και δοκιμών (testing) χρησιμοποιώντας τη train_test_split λειτουργία από το sklearn. Το σετ δοκιμών θα χρησιμοποιηθεί αργότερα για την αξιολόγηση της απόδοσης του

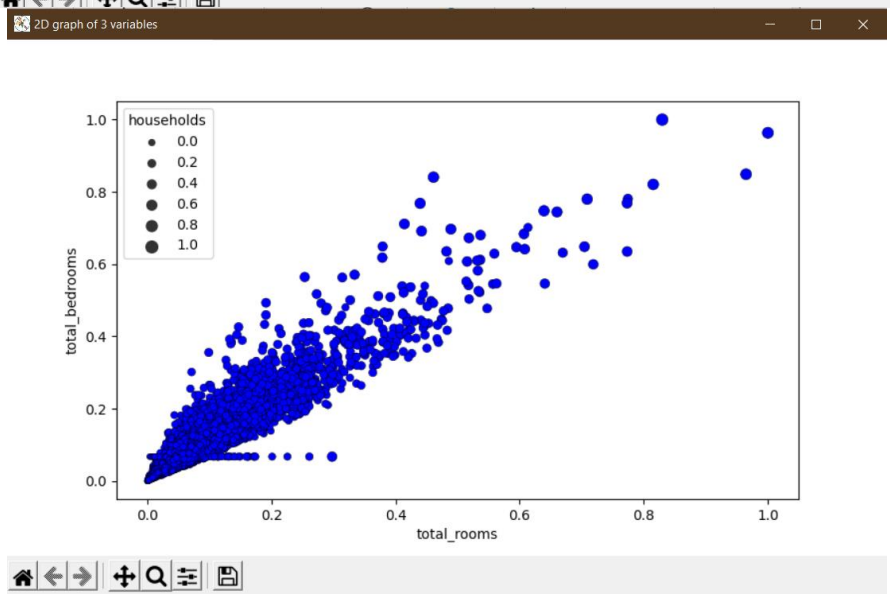
εκπαιδευμένο μοντέλο. Στην συνέχεια, χρησιμοποιούμε την συνάρτηση ColumnTransformer για να σχηματίσουμε έναν ενιαίο χώρο χαρακτηριστικών και καθορίζουμε ποιες στήλες των δεδομένων εισόδου πρέπει να υποβάλλονται σε επεξεργασία. Οι αριθμητικές στήλες επεξεργάζονται από την κλάση SimpleImputer χρησιμοποιώντας την διάμεση τιμή του χαρακτηριστικού σε περίπτωση κενής τιμής, ενώ στην περίπτωση του κατηγορικού χαρακτηριστικού (ocean_proximity) χρησιμοποιείται η κωδικοποίηση OneHotEncoder για να μετατρέψει τα δεδομένα σε αριθμητική/διανυσματική αναπαράσταση. Δημιουργούμε το τελικό pipeline το οποίο περιλαμβάνει δύο στάδια επεξεργασίας δεδομένων. Στο πρώτο στάδιο, οι μεταβλητές εισόδου μετασχηματίζονται με χρήση του αντικειμένου "preprocessor", το οποίο έχει οριστεί προηγουμένως. Ενώ στο δεύτερο στάδιο, οι επεξεργασμένες μεταβλητές εισόδου περνούν στο αντικείμενο "regressor", το οποίο είναι ένα αντικείμενο MLPRegressor με κρυφά επίπεδα (hidden_layer_sizes) (50,100,50) και μέγιστο αριθμό επαναλήψεων (max_iter) ίσο με 500. Το αντικείμενο MLPRegressor εκτελεί και τη μη γραμμική παλινδρόμηση (non-linear regression). Τέλος εκπαιδεύουμε το νευρωνικό δίκτυο (fit) και αξιολογούμε την απόδοση του εκπαιδευμένου μοντέλου με το σετ δοκιμών χρησιμοποιώντας την (score()). Το μοντέλο λόγο του πλήθους των δεδομένων μπορεί να πάρει και 1 λεπτό για να υπολογίσει τελικά την απόδοση του.

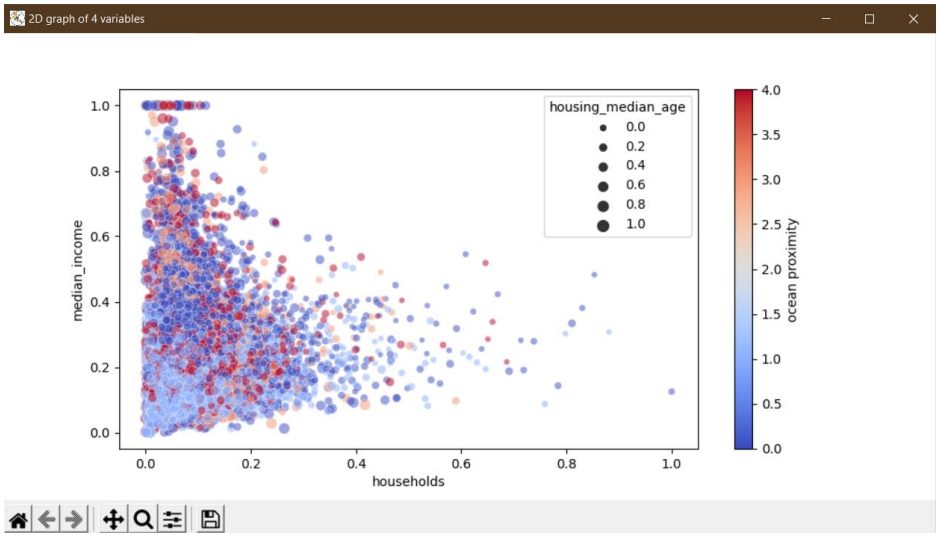
Ενδεικτικά screenshots λειτουργείας

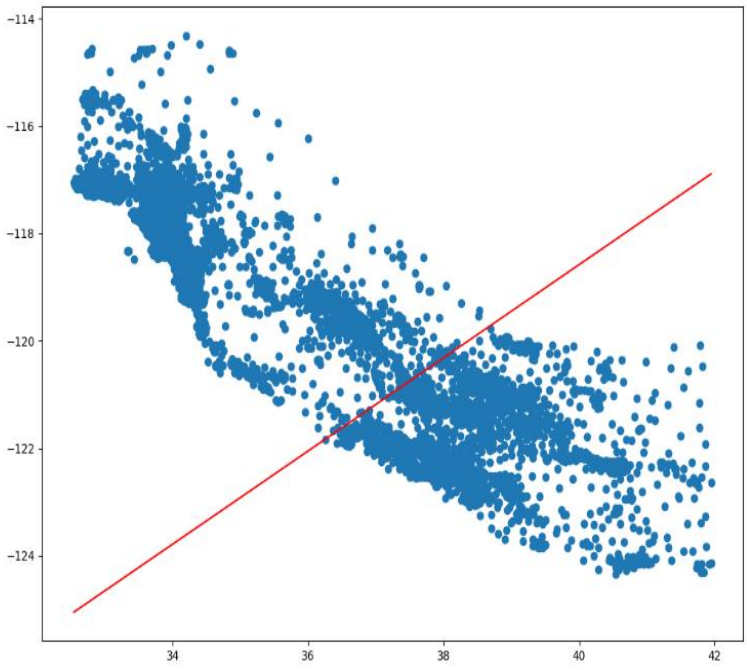



Categorical feature
Original Data
Encoded Data
count
Count
ocean_proximity
NEAR BAY
<1H OCEAN
INLAND
NEAR OCEAN
ISLAND


2D graph of 2 variables
latitude
longitude
median_income
population


2D graph of 3 variables
households
0.0
0.2
0.4
0.6
0.8
1.0
total_bedrooms
total_rooms






```
Average test MSE: 311979.9337267062
Average test MAE: 149.08468535561036

Process finished with exit code 0
```

```
Mean training score:  0.6965941011362698

Process finished with exit code 0
```