Final Project

Μηχανική Μάθηση
2025 - 2026

Ονοματεπώνυμο: Ευθυμιάδης Κωνσταντίνος
ΑΕΜ: 240

Περιεχόμενα

Αρχικό Στάδιο

Αρχικά στο πρώτο μέρος έγινε εισαγωγή των δεδομένων. Τα αρχεία δεδομένων, τα οποία προέρχονται από την ιστοσελίδα του διαγωνισμού, συμπιέστηκαν σε ένα αρχείο τύπου zip με όνομα data.zip. Εντός του αρχείου zip υπάρχουν τα αρχεία:

1. **train_rates_gt.csv**
2. **feature_value_descriptions.csv**
3. **test_hh_features.csv**
4. **feature_descriptions.csv**
5. **train_hh_gt.csv**
6. **train_hh_features.csv**

Δημιουργήθηκαν το dataframe **df_features**, με βάση το αρχείο **train_hh_features.csv** και το dataframe **df_targets**, με βάση το αρχείο **train_hh_gt.csv**. Στη συνέχεια. πραγματοποιήθηκε union αυτών των δύο dataframes με βάση το **hhid** και **survey_id**, δημιουργώντας το dataframe **df**.

Για την διευκόλυνση της ανάλυσης των δεδομένων, δημιουργήθηκε βοηθητικό αρχείο pdf με την χρήση του **ProfileReport**, το οποίο παρέχεται από την βιβλιοθήκη **ydata_profiling**.

Ανάλυση Δεδομένων

Στο dataframe **df** υπάρχουν συνολικά 89 στήλες - features. Πιο συγκεκριμένα:

Πίνακας Δεδομένων

	Όνομα Feature	Περιγραφή
1	**hhid**	Αναπαριστά το ID κάθε νοικοκυριού
2	**com**	Αναπαριστά το ID του μέλους το οποίο έδωσε τις απαντήσεις για την έρευνα
3	**survey_id**	Αναπαριστά το ID της έρευνας
4	**weight**	Εμφανίζει πόσα νοικοκυριά σαν το συγκεκριμένο στην γραμμή, έχουν τα ίδια χαρακτηριστικά. Για παράδειγμα, αν σε μια γραμμή η τιμή για το weight είναι ίση με 2, τότε σημαίνει ότι οι απαντήσεις αυτού του νοικοκυριού αντιπροσωπεύουν 2 νοικοκυριά στην πραγματικότητα
5	**strata**	Αποτελεί μια τιμή που αντιστοιχεί σε συγκεκριμένες ομάδες του πληθυσμού, οι οποίες ορίστηκαν κατά τον σχεδιασμό της έρευνας. Αυτός είναι ένας τρόπος με τον οποίο διασφαλίζεται η σωστή εκπροσώπηση.
6	**utl_exp_ppp17**	Αντιπροσωπεύει τις ημερήσιες δαπάνες του νοικοκυριού για τις υπηρεσίες κοινής ωφέλειας (π.χ. νερό, ρεύμα)

4	**male**	Δείχνει αν ο αρχηγός του νοικοκυριού είναι άνδρας
8	**hsize**	Εμφανίζει τον αριθμό των ατόμων του νοικοκυριού
9	**num_children5**	Αντιπροσωπεύει τον αριθμό των παιδιών του νοικοκυριού κάτω από 5 ετών
10	**num_children10**	Αντιπροσωπεύει τον αριθμό των παιδιών του νοικοκυριού από 5 έως 10 ετών
11	**num_children18**	Αντιπροσωπεύει τον αριθμό των παιδιών του νοικοκυριού από 10 έως 18 ετών
12	**age**	Εμφανίζει την ηλικία του αρχηγού του νοικοκυριού
13	**owner**	Δείχνει αν το σπίτι στο οποίο μένει το νοικοκυριό είναι δικό τους
14	**water**	Εμφανίζουν αν το οίκημα έχει πρόσβαση σε σύστημα παροχής νερού.
15	**toilet**	Δείχνει αν υπάρχουν εγκαταστάσεις υγιεινής στο οίκημα
16	**sewer**	Εμφανίζει αν η κατοικία είναι συνδεδεμένη με το δημόσιο αποχετευτικό δίκτυο.
17	**elect**	Δηλώνει αν η κατοικία έχει πρόσβαση σε ηλεκτρικό ρεύμα
18	**water_source**	Καταγράφει την κύρια πηγή υδροδότησης του νοικοκυριού. Μπορεί να λάβει συγκεκριμένες τιμές (Συνολικά προσφέρονται 8 διαφορετικες συμβολοσειρές)
19	**sanitation_source**	Περιγράφει το είδος της εγκατάστασης υγιεινής που χρησιμοποιεί το νοικοκυριό. Μπορεί να λάβει συγκεκριμένες τιμές (Συνολικά προσφέρονται 14 διαφορετικές τιμές)
20	**dweltyp**	Εμφανίζει τον τύπο του κτηρίου στον οποίο διανέμει το νοικοκυριό. Μπορεί να λάβει συγκεκριμένες τιμές (Συνολικά προσφέρονται 9 διαφορετικές τιμές)
21	**num_adult_female**	Παρουσιάζει το πλήθος των ενήλικων γυναικών (από 18 έως 69 ετών) στο νοικοκυριό
22	**num_adult_male**	Παρουσιάζει το πλήθος των ενήλικων ανδρών (από 18 έως 69 ετών) στο νοικοκυριό
23	**num_elderly**	Παρουσιάζει το πλήθος των ηλικιωμένων ατόμων στο νοικοκυριό (από 70 ετών και άνω)
24	**employed**	Εμφανίζει αν ο αρχηγός του νοικοκυριού εργάζεται
25	**sworkershh**	Εμφανίζει το ποσοστό των εργαζομένων ενηλίκων στο νοικοκυριό
26	**share_secondary**	Εμφανίζει το ποσοστό των ενηλίκων στο νοικοκυριό που έχουν ολοκληρώσει τη δευτεροβάθμια εκπαίδευση

27	**educ_max**	Παρουσιάζει το ανώτερο βαθμό εκπαίδευσης από τα μέλη του νοικοκυριού. Μπορεί να λάβει συγκεκριμένες τιμές (Συνολικά προσφέρονται 7 τιμές)
28	**sfworkershh**	Εμφανίζει το ποσοστό των ενήλικων που εργάζονται σε επίσημο φορέα
29	**any_nonargic**	Παρουσιάζει αν υπάρχει έστω και ένα μέλος του νοικοκυριού που εργάζεται σε μη-αγροτικό τομέα
30	**sector1d**	Παρουσιάζει τον τομέα απασχόλησης του αρχηγού του νοικοκυριού. Μπορεί να λάβει συγκεκριμένες τιμές (Συνολικά προσφέρονται 17 τιμές)
31	**region_{n}**	Δείχνουν την γεωγραφική ζώνη στην οποία ανήκει το νοικοκυριό.
32	**urban**	Δείχνει αν το νοικοκυριό βρίσκεται σε αστική ή αγροτική περιοχή. Λαμβάνει δύο τιμές
33	**consumed{n}**	Δείχνουν αν το νοικοκυριό έχει καταναλώσει τα είδη που αντιπροσωπεύει η μεταβλητή. Κάθε μια μεταβλητή αντιπροσωπεύει σε ένα συγκεκριμένο προϊόν. Για παράδειγμα η consumed100 αντιπροσωπεύει την κατανάλωση ψωμιού.
34	**cons_ppp17**	Αποτελεί το target στα δεδομένα, η οποία εμφανίζει την ημερήσια κατά κεφαλή κατανάλωση.

Χειρισμός Δεδομένων

Στήλες hhid, com και survey_id

Αρχικά, οι στήλες **hhid**, **com** και **survey_id** διαγράφηκαν καθώς δεν προσφέρουν κάποια χρήσιμη πληροφορία, αφού αποτελούν αναγνωριστικά

Στήλες sector1d και employed

Δεδομένου ότι το **sector1d** περιέχει πληροφορία σχετικά με τον τομέα στον οποίο δουλεύει ο αρχηγός του νοικοκυριού και το **employed** περιέχει την πληροφορία σχετικά με το αν δουλεύει ή όχι ο αρχηγός του νοικοκυριού, μπορούμε να συμπεριλάβουμε την πληροφορία του **employed** στο **sector1d**.
Παρατηρείται ότι η τιμή στο **sector1d** δεν είναι ίση με null μόνο όταν η τιμή στην μεταβλητή **employed** είναι ίση με "Employed", κάτι απολύτως λογικό. Επειδή δεν υπάρχει κάποια τιμή για το **sector1d** στις περιπτώσεις που ο αρχηγός του νοικοκυριού δεν δουλεύει, είναι λογικό να μένουν ως null.
Επομένως, θα γίνει αντικατάσταση ώστε όταν η τιμή του **employed** είναι ίση με "Not Employed" να προστίθεται η νέα τιμή "Unemployed" στην στήλη του **sector1d**.

Η στήλη **employed** πρέπει να διαγραφεί, δεδομένου ότι όλη η πληροφορία της απορροφήθηκε στην στήλη **sector1d**

Χειρισμός missing values

Στα δεδομένα του training παρατηρούνται missing values στις στήλες:
dweltyp, **utl_exp_ppp17**, **sector1d**, **educ_max, share_secondary** και σε όλες τις στήλες **consumed{n}**

- Για τις στήλες **dweltyp**, **sector1d**, **educ_max** και **consumed{n}**, επειδή αποτελούν στήλες με categorical δεδομένα, αντικαταστάθηκαν οι κενές τιμές τους με βάση την επικρατέστερη τιμή σύμφωνα με το **region** που βρισκόντουσαν καθώς και με το **strata** τους.
- Για την στήλη **utl_exp_ppp17**, οι κενές τιμές αντικαταστάθηκαν με τον median, με βάση το **strata** και το **region** στο οποίο ανήκει το νοικοκυριό
- Για τον χειρισμό των κενών τιμών στην στήλη **share_secondary**, ακολουθήθηκε η παρακάτω λογική.
  Επειδή η στήλη **educ_max** δείχνει το υψηλότερο επίπεδο εκπαίδευσης που έχει επιτευχθεί από τουλάχιστον ένα μέλος του νοικοκυριού, τότε αν αυτό είναι κάτω από το "Complete Secondary Education", τότε η τιμή του **share_secondary** θα είναι ίση με 0, αφού κανένα μέλος (και άρα ενήλικας), δεν έχει λάβει αυτήν την εκπαίδευση.
  Για τις υπόλοιπες κενές τιμές που έμειναν, αντικαταστάθηκαν με τον median, με βάση το **strata** και το **region** στο οποίο ανήκει το νοικοκυριό

Μετατροπή των Categorical Variables σε Numerical Variables

Μεταβλητή strata

Η μεταβλητή **strata** πρέπει να χειριστεί ως categorical και όχι ως numeric, καθώς αποτελεί μια ετικέτα και όχι μια ποσοτική μεταβλητή.
Για παράδειγμα, οι τιμές στο strata (π.χ. 1, 2, 3) δεν αντιπροσωπεύουν μεγέθη στα οποία μπορούν να πραγματοποιηθούν μαθηματικές πράξεις, αλλά οριοθετούν διαφορετικά δειγματοληπτικά πλαίσια με ιδιαίτερα χαρακτηριστικά.
Συνεπώς, η μετατροπή τους σε κατηγορικές μεταβλητές επιτρέπει στο μοντέλο να αναγνωρίσει κάθε στρώμα ως ανεξάρτητη ομάδα.

Εφαρμογή One-Hot-Encoding για όλες τις στήλες των categorical variables

Για την μετατροπή των categorical μεταβλητών σε numeric χρησιμοποιήθηκε η τεχνική **One-Hot-Encoding**, κατά την οποία κάθε κατηγορία αναπαρίσταται από μια δυαδική μεταβλητή.

Διαχωρισμός Δεδομένων σε Σύνολο Εκπαίδευσης και Σύνολο Ελέγχου

Για τον διαχωρισμό των δεδομένων, απομονώθηκαν στο dataframe **features** όλες οι στήλες του dataframe **df**, εκτός από την στήλη **cons_ppp17**, η οποία αποτελεί το target, και την στήλη **weight**, η οποία περιέχει τα βάρη κάθε γραμμής. Επιπλέον, δημιουργήθηκε το dataframe **target**, με την στήλη **cons_pp_17** και το dataframe **weigths**, με την στήλη **weight**.

Για τον διαχωρισμό των δεδομένων χρησιμοποιήθηκε η συνάρτηση **train_test_split**, η οποία παρέχεται από την βιβλιοθήκη της **sklearn**. Τα δεδομένα χωρίστηκαν σε ποσοστό 80% train και 20% test. Χρησιμοποιήθηκε seed ίσο με 42.

Αλγόριθμοι Μηχανικής Μάθησης

Στα πλαίσια της εργασίας αναπτύχθηκαν συνολικά 3 αλγόριθμοι μηχανικής μάθησης και ένας αλγόριθμος Βαθιάς Μάθησης. Πιο συγκεκριμένα:

Αλγόριθμοι Μηχανικής Μάθησης

Random Forest

Ως πρώτος αλγόριθμος μηχανικής μάθησης, επιλέχθηκε ο RandomForestRegressor. Το μοντέλο δημιουργήθηκε σύμφωνα με το παρακάτω:

```
random_forest_model= RandomForestRegressor(
   n_estimators= 200,
   min_samples_leaf= 5,
   max_features= 0.5,
   max_depth= 10,
   random_state=seed
)
```

Χρησιμοποιήθηκαν συνολικά 200 δέντρα. Τέθηκε max_depth ίσο με 10, καθώς και max_feature= 0.5 και min_samples_leaf= 5, με σκοπό την μείωση του overfit.

Στην συνέχεια η εκπαίδευση του μοντέλου πραγματοποιήθηκε με την ενσωμάτωση των βαρών, μέσω της παραμέτρου sample_weight, με σκοπό την απόδοση της κατάλληλης βαρύτητας σε κάθε νοικοκυριό, ανάλογα με την αντιπροσωπευτικότητά του στον συνολικό πληθυσμό.

```
random_forest_model.fit(x_train, y_train, sample_weight=w_train)
```

Αξιολόγηση

Για την αξιολόγηση χρησιμοποιήθηκαν οι μετρικές MAE και MAPE. Και οι δύο μετρικές υπολογίστηκαν ως σταθμισμένες, ενσωματώνοντας τα βάρη από τα δεδομένα.

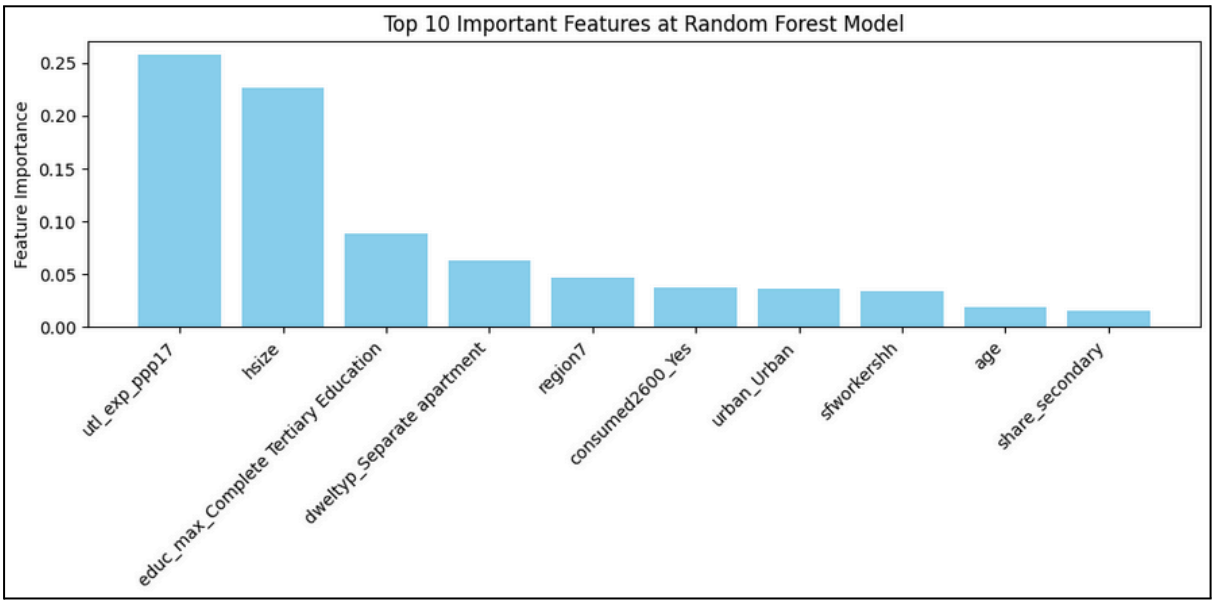
Τα αποτελέσματα ήταν τα εξής:

```
MAE for train is:  2.929601278300621
MAE for test is:  3.2529337281368482
```

```
MAPE for train is:  0.3058683370237525
MAPE for test is:  0.33469783304042217
```

Σημαντικότητα Μεταβλητών

Για το Random Forest μοντέλο, οι 10 πιο σημαντικές μεταβλητές είναι οι παρακάτω:



Lasso Regression

Ως δεύτερος αλγόριθμος μηχανικής μάθησης, επιλέχθηκε ο Lasso. Επιλέχθηκε λόγω της ιδιότητάς του να πραγματοποιεί αυτόματη επιλογή χαρακτηριστικών, μειώνοντας την πολυπλοκότητα του προβλήματος.

Πρώτα γίνεται scaling στα δεδομένα για να έρθουν όλες οι μεταβλητές στην ίδια κλίμακα, ώστε η ποινή του Lasso να μην επηρεάζεται από το μέγεθος των τιμών τους.

Χρησιμοποιήθηκε το LassoCV με σκοπό την εύρεση της καλύτερης τιμής alpha. Πιο συγκεκριμένα:

```
lasso_model = LassoCV(
    cv= 5,
    max_iter= 20000,
    n_jobs= -1,
    random_state= seed
)
```

Η εκπαίδευση του μοντέλου πραγματοποιήθηκε με την ενσωμάτωση των βαρών, μέσω της παραμέτρου sample_weight, με σκοπό την απόδοση της κατάλληλης βαρύτητας σε κάθε νοικοκυριό, ανάλογα με την αντιπροσωπευτικότητά του στον συνολικό πληθυσμό.

```
lasso_model.fit(x_train_scaled, y_train, sample_weight=w_train)
```

Η τιμή του alpha που επιλέχθηκε είναι η παρακάτω:

```
Best alpha: 0.004993713831951244
```

Συνολικά το lasso κράτησε συνολικά 111 features από τα 141

Αξιολόγηση

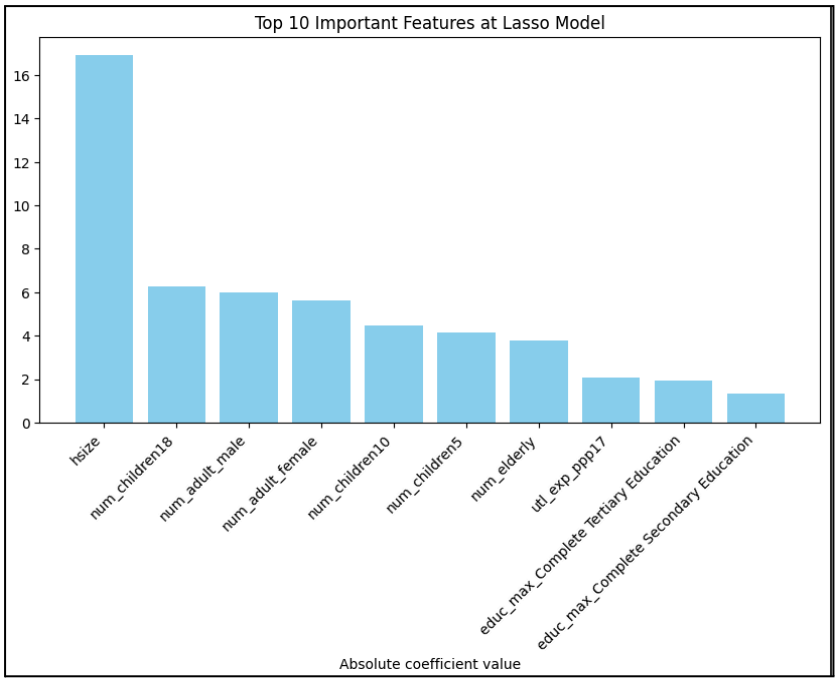
Για την αξιολόγηση χρησιμοποιήθηκαν οι μετρικές MAE και MAPE. Και οι δύο μετρικές υπολογίστηκαν ως σταθμισμένες, ενσωματώνοντας τα βάρη από τα δεδομένα.

Τα αποτελέσματα ήταν τα εξής:

```
MAE for train is:  3.735607358636518
MAE for test is:  3.724409004493734

MAPE for train is:  0.4119655327604145
MAPE for test is:  0.4119711405730425
```

Σημαντικότητα Μεταβλητών

Για το Lasso μοντέλο, οι 10 πιο σημαντικές μεταβλητές είναι οι παρακάτω:



XGBRegressor

Ως τρίτος αλγόριθμος μηχανικής μάθησης, επιλέχθηκε ο xgb. Επιλέχθηκε διότι, σε αντίθεση με τα δέντρα του Random Forest τα οποία είναι ανεξάρτητα μεταξύ τους, κάθε νέο δέντρο προσπαθεί να διορθώσει τα σφάλματα των προηγουμένων.

Οι παράμετροι χρησιμοποιήθηκαν με γνώμονα την αποφυγή του overfitting και την ενίσχυση της σταθερότητας του μοντέλου, επιτρέποντας στον αλγόριθμο να μάθει τις γενικές τάσεις των δεδομένων

```
xgb_model= XGBRegressor(
   n_estimators=1500,
   learning_rate=0.05,
   max_depth=3,
   min_child_weight=5,
   random_state= seed
)
```

- ➢ Επιλέχθηκε μεγάλος αριθμός δέντρων ώστε το μοντέλο να μάθει σταδιακά πολύπλοκες σχέσεις στα δεδομένα
- ➢ Χαμηλή τιμή learning_rate, ώστε κάθε νέο δέντρο να συνεισφέρει λίγο στο τελικό αποτέλεσμα, κάνοντας τη μάθηση πιο σταθερή και μειώνοντας το overfitting.
- ➢ δημιουργία ρηχών δέντρων με την χρήση του max_depth, για απλούστερο μοντέλο που γενικεύει καλύτερα
- ➢ Με την επιλογή της χαμηλής τιμής του min_child_weight αποτρέπεται η δημιουργία φύλλων που βασίζονται σε πολύ λίγα δεδομένα, μειώνοντας την επίδραση θορύβου

Στην συνέχεια, η εκπαίδευση του μοντέλου πραγματοποιήθηκε με την ενσωμάτωση των βαρών, μέσω της παραμέτρου sample_weight, με σκοπό την απόδοση της κατάλληλης βαρύτητας σε κάθε νοικοκυριό, ανάλογα με την αντιπροσωπευτικότητά του στον συνολικό πληθυσμό.

```
xgb_model.fit(x_train, y_train, sample_weight=w_train)
```

Αξιολόγηση

Για την αξιολόγηση χρησιμοποιήθηκαν οι μετρικές MAE και MAPE. Και οι δύο μετρικές υπολογίστηκαν ως σταθμισμένες, ενσωματώνοντας τα βάρη από τα δεδομένα.

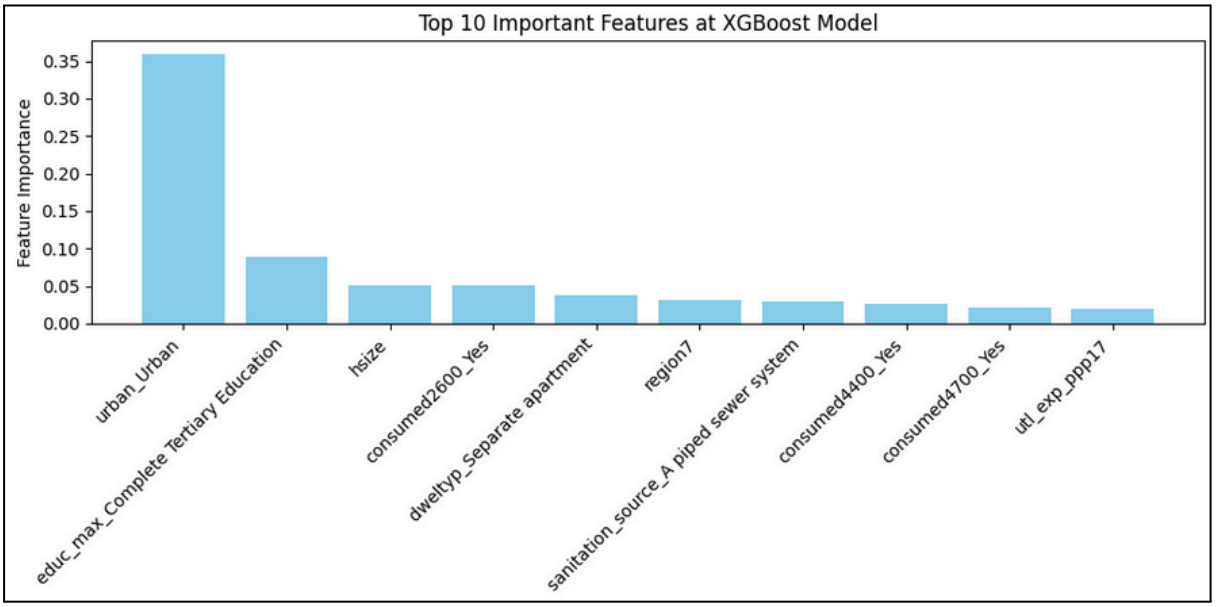
Τα αποτελέσματα ήταν τα εξής:

```
MAE for train is:  2.897513352373098
MAE for test is:  3.091293794819511
```

```
MAPE for train is:  0.2834934726039908
MAPE for test is:  0.2994900862268782
```

Σημαντικότητα Μεταβλητών

Για το xgb μοντέλο, οι 10 πιο σημαντικές μεταβλητές είναι οι παρακάτω:



Αλγόριθμος Βαθιάς Μάθησης - MLP

Για την επίλυση του προβλήματος επιλέχθηκε ένα Multi-Layer Perception (MLP).

Πριν την εκπαίδευση του μοντέλου είναι αναγκαία η χρήση Scaling στα δεδομένα εκπαίδευσης.

Το μοντέλο δημιουργήθηκε με βάση το παρακάτω:

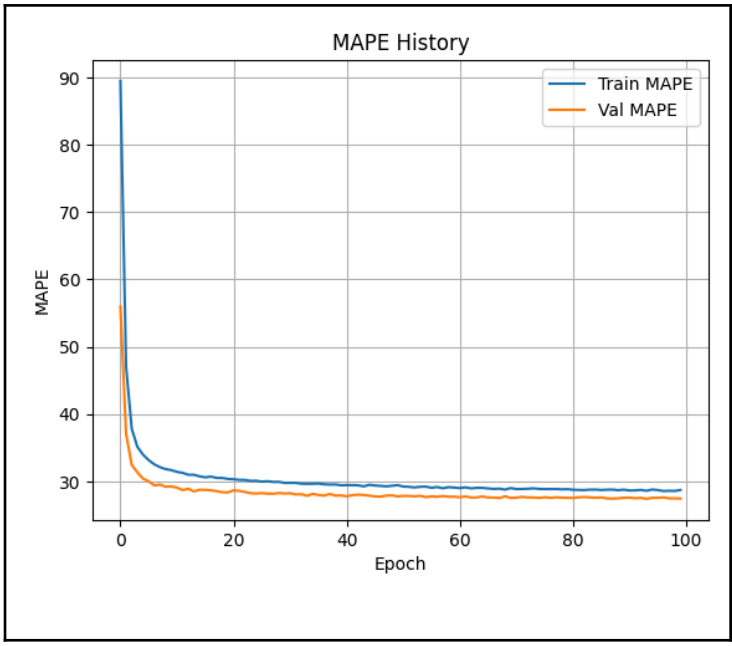
```
input_size= (x_train_scaled.shape[1], )

mlp_model = tf.keras.Sequential([
   tf.keras.layers.Input(shape= input_size),

   tf.keras.layers.Dense(units=128, activation='relu', use_bias=True),
   tf.keras.layers.BatchNormalization(),
   tf.keras.layers.Dropout(0.3),

   tf.keras.layers.Dense(units=64, activation='relu', use_bias=True),
   tf.keras.layers.BatchNormalization(),
```

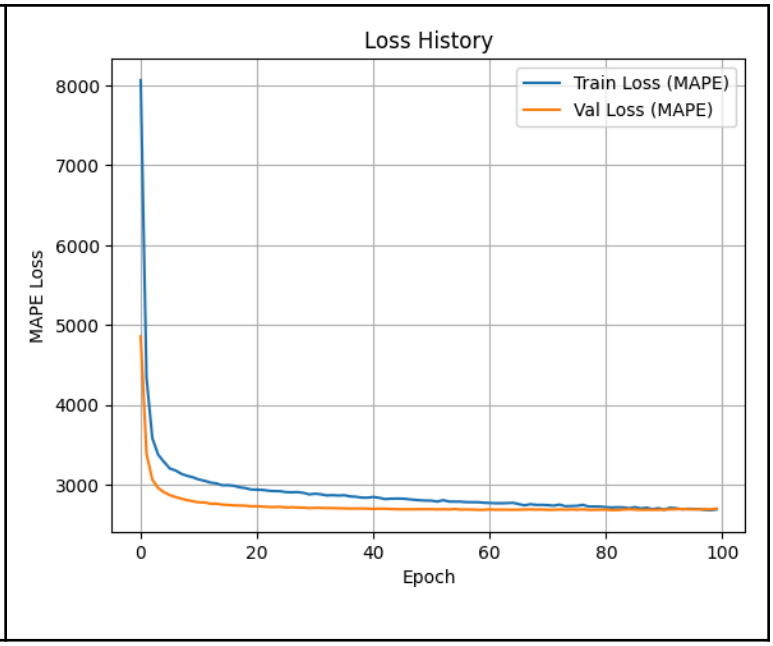
```
    tf.keras.layers.Dropout(0.2),

    tf.keras.layers.Dense(32, activation='relu'),

    tf.keras.layers.Dense(1)
], name= "MLP")
```

1. Πρώτο layer (Layer Εισόδου): Το μέγεθος του input αντιστοιχεί με στον αριθμό των χαρακτηριστικών του dataset.
2. Πρώτο Hidden Layer: Χρησιμοποιούνται 128 νευρώνες, με activation συνάρτηση την ReLU, με σκοπό το μοντέλο να μάθει τις πολύπλοκες σχέσεις στα δεδομένα
   2.1. Batch Normalization: Εφαρμόζεται με σκοπό την βελτίωση της σύγκλισης του μοντέλου
   2.2. DropOut: Πραγματοποιείται dropOut με σκοπό την μείωση του overfitting, απενεργοποιώντας τυχαία το 30% των νευρώνων του layer κατά την εκπαίδευση.
3. Δεύτερο Hidden Layer: Χρησιμοποιούνται 64 νευρώνες, με activation συνάρτηση την ReLU, μειώνοντας σταδιακά την διαστασιμότητα.
   3.1. Αντίστοιχα Πραγματοποιείται Batch Normalization και έπειτα DropOut σε βαθμό όμως τώρα 20%
4. Τρίτο Hidden Layer: Χρησιμοποιούνται 32 νερώνες, με activation συνάρτηση την ReLU.
5. Layer Εξόδου: Αποτελείται από έναν νευρώνα χωρίς κάποια activation συνάρτηση, καθώς πρόκειται για πρόβλημα regression, με αποτέλεσμα η έξοδος να είναι μια συνεχής τιμή

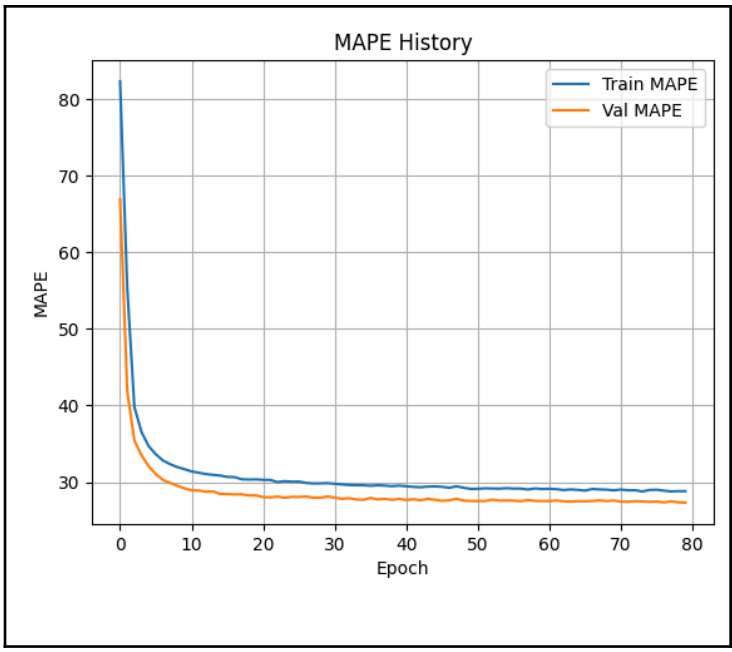
Εκπαίδευση μοντέλου

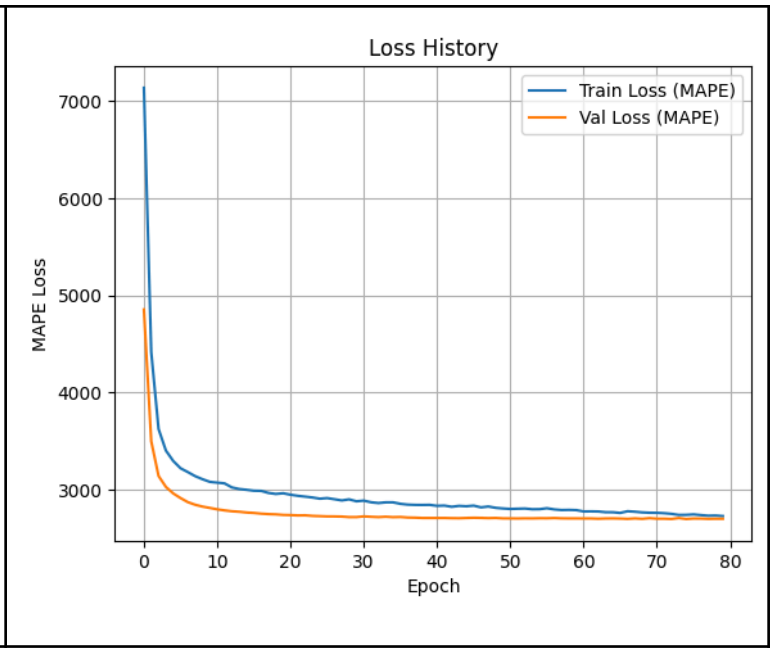
Για την εκπαίδευση του μοντέλου χρησιμοποιήθηκε ο optimizer Adam με learning rate= 0.0001, επιτρέποντας σταθερή και ομαλή σύγκλιση χωρίς απότομες μεταβολές στο loss. Ως Loss Function επιλέχθηκε το MAE, ενώ ως μέτρο αξιολόγηση επιλέχθηκε η MAPE.
Το batch size ορίστηκε ίσο με 128, ενώ αρχικά προγραμματίστηκαν 100 epochs.
Τα learning curves που προέκυψαν (για MAPE και loss) είναι τα ακόλουθα:





Έπειτα από παρατήρηση των curves διαπιστώθηκε ότι μετά τα 80 epochs το μοντέλο MLP είναι επαρκώς εκπαιδευμένο χωρίς να χρειάζεται να σπαταληθούν περισσότεροι πόροι σε επαναλήψεις, ενώ επιπλέον μειώνεται ο κίνδυνος της υπερ-προσαρμογής.
Επομένως το μοντέλο εκπαιδεύτηκε εκ νέου για 80 epochs. Τα learning curves που προέκυψαν είναι τα ακόλουθα





Χειρισμός αρχείου test_hh_features.csv

Το test_hh_features.csv, το οποίο αποτελεί το αρχείο που με βάση αυτό θα γίνουν οι προβλέψεις για τον διαγωνισμό, αποθηκεύεται στο dataframe **test_features**.

Εντός του **test_features** οι μεταβλητές **employed** και **sector1d** χειρίζονται με τον αντίστοιχο τρόπο που έγινε και στα πλαίσια της ανάλυσης των δεδομένων εκπαίδευσης, που στην συνέχεια διαγράφεται η **employed**.

Έπειτα, τα missing values χειρίζονται με αντίστοιχο τρόπο με το στάδιο της ανάλυσης των δεδομένων εκπαίδευσης. Στην συνέχεια όλες οι categorical μεταβλητές μετατρέπονται σε numeric, με την χρήση της μεθόδου One-Hot-Encoding.

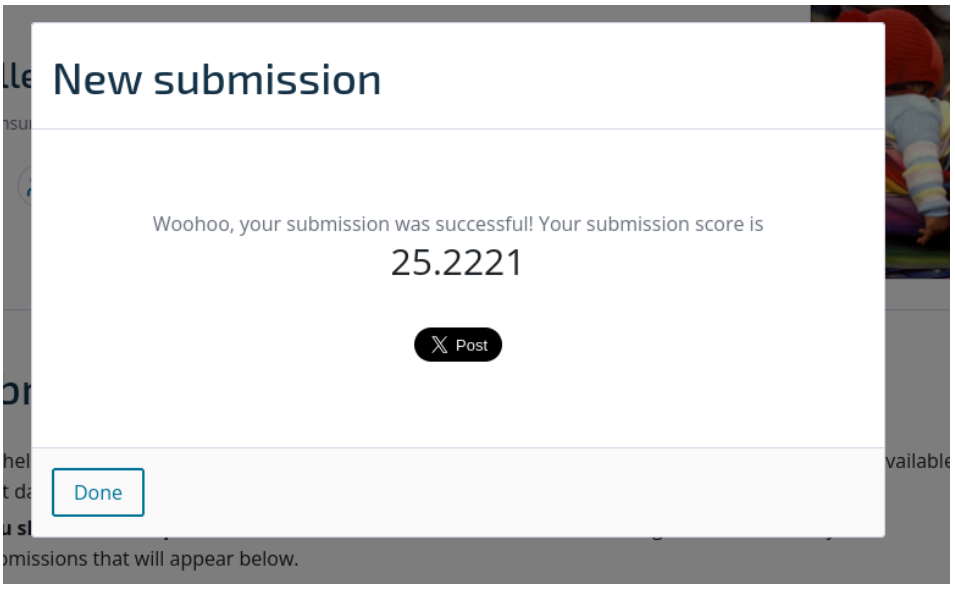
Πρόβλεψη

Για την πρόβλεψη αφαιρούνται οι στήλες **weight**, **hhid**, **com** και **survey_id** και δημιουργείται ένα νέο dataframe με όνομα **features**. Έπειτα, για κάθε ένα μοντέλο που δημιουργήθηκε, πραγματοποιείται πρόβλεψη με βάση τα δεδομένα του dataframe **features**.
Τέλος, δημιουργούνται τα δύο αρχεία που απαιτούνται για τον διαγωνισμό, με βάση τα αποτελέσματα των προβλέψεων

Αποτελέσματα Διαγωνισμού

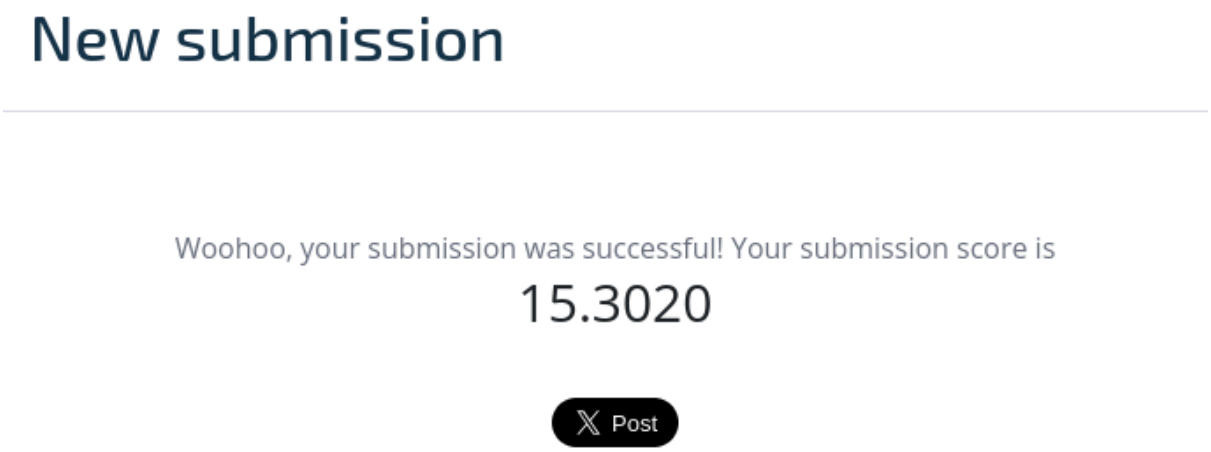
Μοντέλο Random Forest

Το συγκεκριμένο Random Forest Regressor μοντέλο συνολικά συγκέντρωσε score= 25.2221



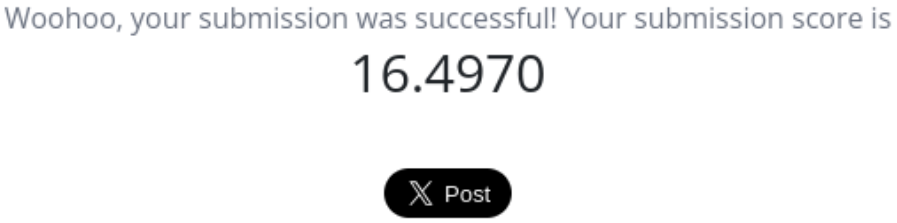
Μοντέλο Lasso

Το συγκεκριμένο Lasso μοντέλο συνολικά συγκέντρωσε score= 15.3020



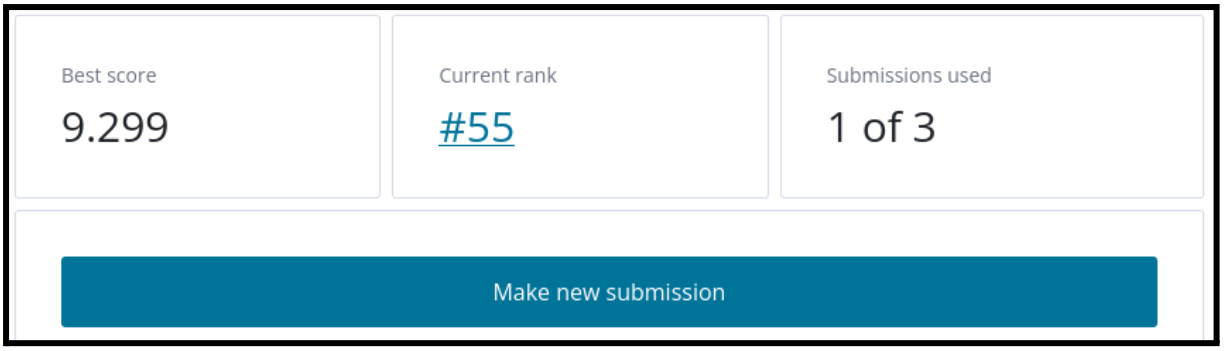
Μοντέλο XGBRegressor

Το συγκεκριμένο xgb μοντέλο συνολικά συγκέντρωσε score= 16.4970



Μοντέλο MLP

Το συγκεκριμένο mlp μοντέλο συνολικά συγκέντρωσε score= 9.299, αποτελώντας και το καλύτερο μοντέλο σε σχέση με τα άλλα. Με την χρήση αυτού του μοντέλου η θέση που έλαβα στο leaderboard ήταν 55η.



Σχολιασμός

Από τα αποτελέσματα του διαγωνισμού φαίνεται ότι το **MLP** αποτελεί το πιο αποτελεσματικό μοντέλο, ακολουθούμενο από το **Lasso**, το **XGBoost Regressor** και τέλος το **Random Forest Regressor**.

Το **MLP** έχει την ικανότητα να μαθαίνει δύσκολες και περίπλοκες σχέσεις μεταξύ των χαρακτηριστικών και του στόχου, αντίθετα με τα παραδοσιακά μοντέλα τα οποία δυσκολεύονται.
Το **Lasso** λόγω της ικανότητάς του να μειώνει τον θόρυβο και να επιλέγει μόνο τα σημαντικά features κατάφερε να ξεπεράσει τους άλλους δύο αλγορίθμους.
Τα άλλα δύο δεντρικά μοντέλα παρουσιάζουν μειωμένη απόδοση, λόγω πιθανού overfitting.

Γίνεται αντιληπτό, ότι για προβλήματα με μεγάλο όγκο δεδομένων και πολύπλοκες σχέσεις μεταξύ τους, τα νευρωνικά μοντέλα αποδίδουν αρκετά καλύτερα, ενώ τα γραμμικά μοντέλα συνεχίζουν να παραμένουν χρήσιμα.

Τρόπο Βελτίωσης Δεδομένων

Για την βελτίωση των αποτελεσμάτων προτείνονται οι παρακάτω ιδέες.

Η προσθήκη δεδομένων για το καθαρό εισόδημα καθώς και για πιθανά εισοδήματα από ακίνητη περιουσία σε κάθε νοικοκυριό θα βοηθούσε σημαντικά τους αλγορίθμους. Η κατανάλωση από μόνη της μπορεί να είναι παραπλανητική, καθώς μπορεί ένα νοικοκυριό να εμφανίζει υψηλά έξοδα, χωρίς όμως να έχει τους αντίστοιχους πόρους ή αντίθετα να λαμβάνει αρκετά εισοδήματα, αλλά τα έξοδά τους να είναι περιορισμένα.

Η προσθήκη πληροφορίας σχετικά με την ύπαρξη μελών στο νοικοκυριό με χρόνια νοσήματα θα ήταν χρήσιμη πληροφορία. Πολλά νοικοκυριά εμφανίζουν υψηλά έξοδα λόγω ιατρικών αναγκών, ενώ ταυτόχρονα αντιμετωπίζουν σοβαρά προβλήματα διαβίωσης.