ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ

ΠΜΣ: ΤΕΧΝΙΤΗ ΝΟΗΜΟΣΥΝΗ

Τελική Εργασία

Ονοματεπώνυμο: Μπόσινας Παναγιώτης

ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ

Εισαγωγή

Η εργασία αφορά την πρόβλεψη του ποσοστού φτώχιας με βάση την κατανάλωση των νοικοκυριών. Ως δεδομένα αξιοποιήθηκαν διάφορα στοιχεία από έρευνες που σχετίζονται με τις συνθήκες διαβίωσης, το μορφωτικό επίπεδο, την οικογενειακή κατάσταση και την οικονομική και επαγγελματική ευημερία των συμμετεχόντων. Η μελέτη αυτή αποτελεί πρόβλημα παρεμβολής (regression).

Προεπεξεργασία των δεδομένων

Τα δεδομένα περιγράφηκαν με χρήση του Profile Report του ydata profiling. Αποτελούνται από 100000 περίπου δείγματα με 88 χαρακτηριστικά, 14 αριθμητικά, 23 κατηγορικά και 51 Boolean. Το 0.2% των τιμών είναι ελλιπείς ενώ υπάρχουν και χαρακτηριστικά με υψηλό correlation μεταξύ τους.

Ως πρώτο βήμα, αφαιρέθηκαν οι στήλες 'hhid', 'survey_id', 'com' και 'strata' καθώς δεν αποτελούν δεδομένα χρήσιμα για την εκπαίδευση. Στην συνέχεια, εφαρμόστηκε label encoding σε όλες οι Boolean μεταβλητές ώστε να παίρνουν την τιμή 1 όταν είναι True και 0 όταν είναι False. Για τις κατηγορικές μεταβλητές εφαρμόστηκε mapping ώστε να λαμβάνουν ακέραια ψηφία ως διακριτές τιμές. Έπειτα, έγινε αφαίρεση των ελλιπών τιμών.

Επόμενο βήμα κατά την προεπεξεργασία ήταν η μετατροπή των dataframes σε numpy arrays, με x τον πίνακα των χαρακτηριστικών, y τον πίνακα των targets (δλδ. το consumption των νοικοκυριών) και weights τα βάρη της στήλης weight που υποδεικνύει το κατά πόσο το εκάστοτε νοικοκυριό αντιπροσωπεύει τον πραγματικό πληθυσμό. Τα δεδομένα διαχωρίστηκαν σε δεδομένα εκπαίδευσης και επικύρωσης με ποσοστό 90%-10% και στην συνέχεια έγινε z-score κανονικοποίηση με βάση τα training data με χρήση του StandardScaler().

Εκπαίδευση των μοντέλων

Δοκιμάστηκαν τρία μοντέλα μηχανικής μάθησης και ένα μοντέλο βαθιάς μάθησης. Συγκεκριμένα, έγινε σύγκριση μεταξύ των επιδόσεων ενός Linear Regressor με Lasso ομαλοποίηση, ενός Random Forest, του XGBoost αλγορίθμου και ενός MLP νευρωνικού δικτύου.

ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ

Παρακάτω φαίνεται η εκπαίδευση των αλγορίθμων και οι παράμετροι που επιλέχθηκαν:

```
lasso_reg = Lasso()
lasso_reg.fit(x_train, y_train, sample_weight=w_train)
```

```
rf_reg =
RandomForestRegressor(n_estimators=100,max_depth=20,min_samples_split=10,random_sta
te=seed,n_jobs=-1)
rf_reg.fit(x_train, y_train, sample_weight=w_train)
```

```
xgb_reg =
xgb.XGBRegressor(n_estimators=300,learning_rate=0.05,max_depth=8,random_state=seed,
n_jobs=-1)
xgb_reg.fit(x_train, y_train, sample_weight=w_train)
```

```
def create_nn_model(input_dim):
    model = keras.Sequential([
        layers.Dense(256, activation='relu', input_shape=(input_dim,)),
        layers.Dropout(0.3),
        layers.Dense(128, activation='relu'),
        layers.Dense(64, activation='relu'),
        layers.Dropout(0.2),
        layers.Dense(32, activation='relu'),
        layers.Dense(1)
    ])
    model.compile(optimizer=keras.optimizers.Adam(learning_rate=0.001),
                  loss='mse', metrics=['mae'])
    model.summary()
    return model

nn_model = create_nn_model(x_train.shape[1])
history = nn_model.fit(x_train,
y_train,sample_weight=w_train,epochs=50,batch_size=32)
```

Η ακρίβεια για το validation set εκτιμήθηκε με το Mean Absolute Error ως μετρική:

Πίνακας 1: Σύγκριση των επιδόσεων των διάφορων αλγορίθμων παλινδρόμησης

Αλγόριθμος	Validation set MAE
Lasso Linear Regressor	3.96
Random Forest Regressor	3.26
XGBoost Regressor	3.09
Multi Layered Perceptron	3.18

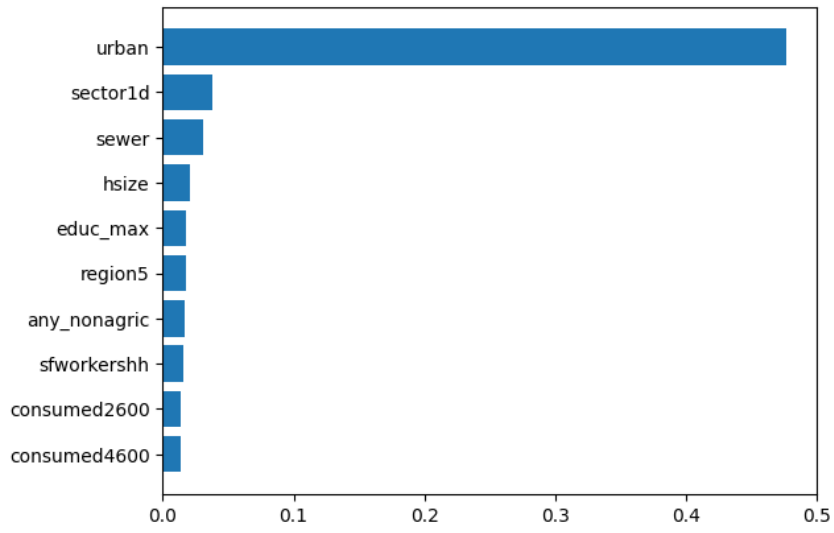
Είναι εμφανές ότι ο XGBoost είναι ο καλύτερος αλγόριθμος. Επομένως, έγινε παραμετρική ανάλυση του αλγορίθμου με σκοπό να βρεθούν οι βέλτιστες υπερπαράμετροι. Για τον σκοπό αυτό εφαρμόστηκε Gri Search με 3-fold cross validation:

```
disp = ConfusionMatrixDisplay(confusion_matrix=cm, display_labels=cifar10_labels)
disp.plot(cmap=plt.cm.Blues)
plt.title("Confusion Matrix")
plt.show()



param_grid = {'n_estimators':[100, 200, 400],
              'min_child_weight': [1, 3, 5],
              'max_depth': [3, 5, 7, 8, 10],
              'learning_rate': [0.01, 0.05, 0.1]}

xgb_opt = xgb.XGBRegressor(objective='reg:absoluteerror', random_state=seed)

grid_search = GridSearchCV(estimator=xgb_opt, param_grid=param_grid, cv=3,
scoring='neg_mean_absolute_error', n_jobs=-1)
grid_search.fit(x_train, y_train)
print(f"Best parameters: {grid_search.best_params_}")
print(f"Best score: {grid_search.best_score_}")
best_xgb = grid_search.best_estimator_
```

```
Best parameters: {'learning_rate': 0.05, 'max_depth': 7, 'min_child_weight': 5, 'n_estimators': 400}
Best score: -3.0457646515978545
```

Επομένως, ο βελτιστοποιημένος XGBoost είναι ελαφρώς καλύτερος σε επίδοση.

Με βάση το μοντέλο, τα 10 πιο σημαντικά χαρακτηριστικά είναι τα ακόλουθα:



Εικόνα 1. Ραβδόγραμμα των feature importances

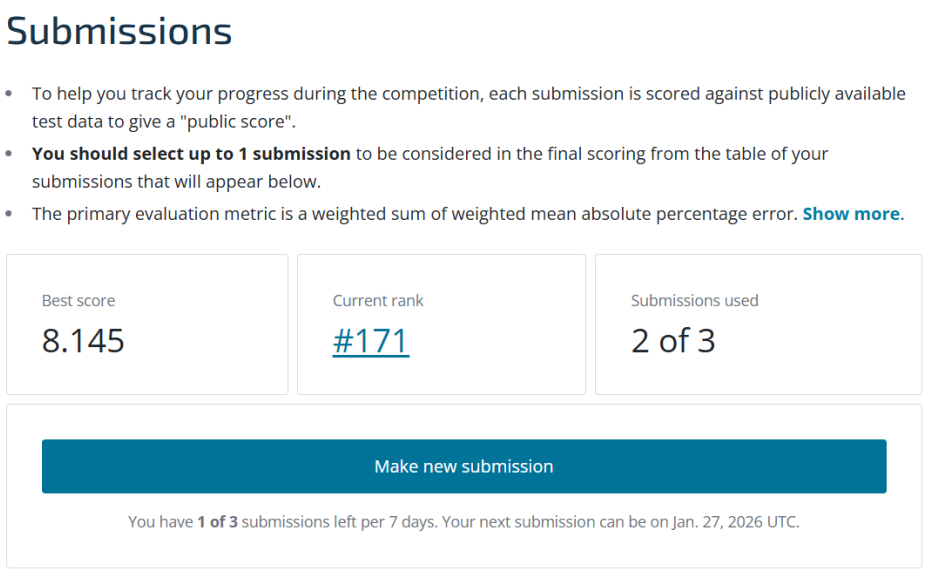
Παρατηρούμε ότι το αν κάποιος ζει σε αστική ή επαρχιακή ζώνη παίζει καθοριστικό ρόλο στην πρόβλεψη της φτώχιας. Άλλα σημαντικά χαρακτηριστικά είναι το ποσοστό των εργαζόμενων μελών του νοικοκυριού, το αν εργάζεται κάποιο μέλος στον αγροτικό τομέα, η περιοχή όπου διαμένουν, το μέγιστο επίπεδο εκπαίδευσης που έχουν και το πλήθος των μελών του νοικοκυριού.

Αξίζει να σημειωθεί ότι δοκιμάστηκε και μείωση της διάστασης των δεδομένων με PCA, αλλά τα αποτελέσματα σε ορισμένες περιπτώσεις ήταν χειρότερα, πιθανώς λόγω απώλειας πληροφορίας.

Πρόβλεψη για τα δεδομένα ελέγχου

Τα δεδομένα ελέγχου χωρίζονταν σε τρία surveys. Έγινε η ίδια προεπεξεργασία που ακολουθήθηκε και για τα train και validation sets και έγινε πρόβλεψη πάνω στον βελτιστοποιημένο XGBoost αλγόριθμο. Τα αποτελέσματα για την κατανάλωση των νοικοκυριών αποθηκεύτηκαν σε .csv αρχείο. Στην συνέχεια έγινε πρόβλεψη των ποσοστών των νοικοκυριών που βρίσκονται κάτω από ένα όριο για κάθε survey με βάση τα poverty thresholds του survey 300000 των train data, όπως ακριβώς ορίζει ο διαγωνισμός. Τα αποτελέσματα αποθηκεύτηκαν και πάλι σε αρχείο .csv.

Το τελικό σκορ που επιτυγχάνεται με βάση το weighted MAPE είναι 8.145.



Γενικά βλέπουμε ότι τα χαρακτηριστικά συσχετίζονται με μη γραμμικές σχέσεις, γεγονός που φαίνεται από την αισθητά καλύτερη απόδοση του XGBoost αλγορίθμου και του νευρωνικού από τον γραμμικής παλινδρόμησης ταξινομητή.