



MACHINE LEARNING

CONCLUSION ANALYSIS

George Krasakis
AP23012

ΠΕΡΙΕΧΟΜΕΝΑ

1. PART 1	2
1.1 Διερεύνηση των δεδομένων (EDA) και προ-επεξεργασία	2
1.2 Τυποποίηση Δεδομένων	4
1.3 Δημιουργία συνόλου εκπαίδευσης και δοκιμής	4
1.4 Ανάπτυξη γραμμικού μοντέλου	5
1.5 Παλινδρόμηση με Random Forest	6
2. PART 2	7
2.1 Κατηγοριοποίηση με πλήρως διασυνδεδεμένο δίκτυο	7
2.2 Κατηγοριοποίηση με ένα συνελικτικό δίκτυο	9
2.3 Πειραματισμός για την ανάπτυξη μοντέλων νευρωνικών δικτύων	11
3. PART 3	14
3.1 Πειραματισμός για την ανάπτυξη μοντέλων νευρωνικών δικτύων	14

1. PART 1

1.1 Διερεύνηση των δεδομένων (EDA) και προ-επεξεργασία

```
Data columns (total 10 columns):  
#   Column                Non-Null Count  Dtype  
---  -  
0    longitude             20640 non-null  float64  
1    latitude               20640 non-null  float64  
2    housing_median_age     20640 non-null  float64  
3    total_rooms             20640 non-null  float64  
4    total_bedrooms         20433 non-null  float64  
5    population              20640 non-null  float64  
6    households              20640 non-null  float64  
7    median_income           20640 non-null  float64  
8    median_house_value      20640 non-null  float64  
9    ocean_proximity         20640 non-null  object  
dtypes: float64(9), object(1)  
memory usage: 1.6+ MB  
None
```

```
<class 'pandas.core.frame.DataFrame'  
RangeIndex: 20640 entries, 0 to 20639
```

Παρατηρήθηκε, ότι υπάρχουν κάποιες τιμές που λείπουν

```
longitude      0  
latitude       0  
housing_median_age  0  
total_rooms    0  
total_bedrooms 207  
population     0  
households     0  
median_income  0  
median_house_value  0  
ocean_proximity  0  
dtype: int64
```

Επίσης, η στήλη **Ocean_proximity**, αποτελεί κατηγορική μεταβλητή και θα πρέπει να κωδικοποιηθεί με τη διαδικασία **One-hot encoding** και προκύπτουν:

```
[ ] print(df['ocean_proximity'].unique())
print(df['ocean_proximity'].value_counts())
```

```
↗ ['NEAR BAY' '<1H OCEAN' 'INLAND' 'NEAR OCEAN' 'ISLAND']
ocean_proximity
<1H OCEAN    9136
INLAND       6551
NEAR OCEAN   2658
NEAR BAY     2298
ISLAND        5
Name: count, dtype: int64
```

```
[ ] df = pd.get_dummies(df, columns=['ocean_proximity'], drop_first=True)
```

```
[ ] print(df)
```

```
↗
```

	longitude	latitude	housing_median_age	total_rooms	total_bedrooms	\
0	-122.23	37.88	41.0	880.0	129.0	
1	-122.22	37.86	21.0	7099.0	1186.0	
2	-122.24	37.85	52.0	1467.0	190.0	
3	-122.25	37.85	52.0	1274.0	235.0	
4	-122.25	37.85	52.0	1627.0	280.0	
...	
20635	-121.09	39.48	25.0	1665.0	374.0	
20636	-121.21	39.49	18.0	697.0	150.0	
20637	-121.22	39.43	17.0	2254.0	485.0	
20638	-121.32	39.43	18.0	1860.0	409.0	
20639	-121.24	39.37	16.0	2785.0	616.0	

	population	households	median_income	median_house_value	\
0	322.0	126.0	8.3252	452600.0	
1	2401.0	1138.0	8.3014	358500.0	
2	496.0	177.0	7.2574	352100.0	
3	558.0	219.0	5.6431	341300.0	
4	565.0	259.0	3.8462	342200.0	
...	
20635	845.0	330.0	1.5603	78100.0	
20636	356.0	114.0	2.5568	77100.0	
20637	1007.0	433.0	1.7000	92300.0	
20638	741.0	349.0	1.8672	84700.0	
20639	1387.0	530.0	2.3886	89400.0	

	ocean_proximity_INLAND	ocean_proximity_ISLAND	\
0	False	False	
1	False	False	
2	False	False	
3	False	False	
4	False	False	
...	
20635	True	False	
20636	True	False	
20637	True	False	
20638	True	False	
20639	True	False	

	ocean_proximity_NEAR BAY	ocean_proximity_NEAR OCEAN	\
0	True	False	
1	True	False	
2	True	False	
3	True	False	
4	True	False	
...	
20635	False	False	
20636	False	False	
20637	False	False	
20638	False	False	
20639	False	False	

4 νέες Columns. Ακόμη υπάρχει και η column **median_house_value**. Η οποία αποτελεί μια εξαρτώμενη μεταβλητή και δεν πρέπει να είναι μαζί με τις ανεξάρτητες.

1.2 Τυποποίηση Δεδομένων

Πραγματοποιείται η τυποποίηση των δεδομένων και γίνεται διαχωρισμός 70% training και 30% test.

```
# Split the dataset into training (70%) and test (30%) sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)
```

1.3 Δημιουργία συνόλου εκπαίδευσης και δοκιμής

Με βάση τα αποτελέσματα της δημιουργίας των συνόλων δοκιμής και εκπαίδευσης επιβεβαιώνω ότι είναι 70-30.

```
[13] # Confirm the shapes of the resulting sets
print(f"X_train shape: {X_train.shape}, y_train shape: {y_train.shape}")
print(f"X_test shape: {X_test.shape}, y_test shape: {y_test.shape}")

# Now X_train, X_test, y_train, y_test are ready for further processing and modeling

# Transform training and test data
X_train_sc = scaler.transform(X_train)
X_test_sc = scaler.transform(X_test)
```

⇒ X_train shape: (14448, 12), y_train shape: (14448,)
X_test shape: (6192, 12), y_test shape: (6192,)

Συνολικά έχουμε:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 20640 entries, 0 to 20639
```

1.4 Ανάπτυξη γραμμικού μοντέλου

Το συμπέρασμα που βγάζω όσον αφορά την τυποποίηση των δεδομένων είναι ότι έχει επιφέρει θετικό αποτέλεσμα στην απόδοση του γραμμικού μοντέλου. Συγκεκριμένα:

- **Μέσο τετραγωνικό σφάλμα (Mean Squared Error - MSE):** Το MSE είναι χαμηλό, που υποδηλώνει ότι οι προβλέψεις του μοντέλου έχουν μικρή απόκλιση από τις πραγματικές τιμές. Αυτό αποτελεί ένα θετικό σημάδι για την ακρίβεια του μοντέλου .
- **R-squared (R^2):** Το R^2 είναι κοντά στη μονάδα (1), που υποδηλώνει ότι οι ανεξάρτητες μεταβλητές εξηγούν καλά την μεταβλητότητα της εξαρτημένης μεταβλητής. Αυτό σημαίνει ότι το μοντέλο είναι ικανοποιητικό στην πρόβλεψη των δεδομένων.

Συνεπώς, η τυποποίηση των δεδομένων (χρησιμοποιώντας την κανονικοποίηση σε εύρος [0, 1]) έχει βελτιώσει την απόδοση του μοντέλου. Οι χαμηλές τιμές MSE και το υψηλό R^2 είναι καλά σημάδια ότι το μοντέλο είναι ικανό να κάνει ακριβείς προβλέψεις για τις εξαρτημένες μεταβλητές βάσει των ανεξάρτητων μεταβλητών που χρησιμοποιούμε.

1.5 Παλινδρόμηση με Random Forest

Η επίδοση του Random Forest σε σύγκριση με το προηγούμενο γραμμικό μοντέλο έχει κάποιες ξεκάθαρες διαφορές, όπως παρατηρείται από τις μετρικές αξιολόγησης (MSE και R-squared):

- **Mean Squared Error (MSE) Για το Random Forest:** Το μέσο τετραγωνικό σφάλμα είναι χαμηλότερο σε σύγκριση με το γραμμικό μοντέλο. Αυτό σημαίνει ότι οι προβλέψεις του Random Forest είναι πιο κοντά στις πραγματικές τιμές στο σύνολο δοκιμής. **Το χαμηλότερο MSE υποδεικνύει καλύτερη ακρίβεια και απόδοση του μοντέλου.**
- **R-squared (R^2) Για το Random Forest:** Το R^2 είναι επίσης υψηλότερο από αυτό του γραμμικού μοντέλου. Αυτό υποδεικνύει ότι το Random Forest εξηγεί καλύτερα την μεταβλητότητα της εξαρτημένης μεταβλητής με βάση τις ανεξάρτητες μεταβλητές. **Το υψηλότερο R^2 υποδεικνύει καλύτερη προσαρμογή του μοντέλου στα δεδομένα.**

Όσον αφορά τη γραμμική επίδοση:

1. **Μη γραμμικότητα και ικανότητα για περίπλοκα δεδομένα:** Το Random Forest είναι ένα **μη γραμμικό μοντέλο** που μπορεί να ανιχνεύσει και να εκμεταλλευτεί μη γραμμικές σχέσεις μεταξύ των μεταβλητών, κάτι που το γραμμικό μοντέλο (όπως η γραμμική παλινδρόμηση) δεν μπορεί πάντα να κάνει αποτελεσματικά.
2. **Ανθεκτικότητα στην υπερεκπαίδευση:** Το Random Forest έχει τη δυνατότητα να διαχειρίζεται καλύτερα την υπερεκπαίδευση σε σύγκριση με πιο απλά γραμμικά μοντέλα, επειδή χρησιμοποιεί πολλά δέντρα αποφάσεων και έτσι εξισορροπεί την πολυπλοκότητα και την απόδοση του μοντέλου.
3. **Βελτίωση της ακρίβειας λόγω συνόλου μοντέλων:** Το Random Forest συνδυάζει πολλά δέντρα αποφάσεων, καθιστώντας τη μέθοδο πιο αξιόπιστη και συνήθως με μεγαλύτερη ακρίβεια σε σχέση με τα απλά γραμμικά μοντέλα.

Έτσι προκύπτει, ότι η διαφορά στην επίδοση του Random Forest σε σχέση με το γραμμικό μοντέλο είναι αποτέλεσμα της ικανότητάς του να αναγνωρίζει και να προβλέπει μη γραμμικές σχέσεις και της δυνατότητάς του να διαχειρίζεται πιο πολύπλοκα δεδομένα.

2. PART 2

2.1 Κατηγοριοποίηση με πλήρως διασυνδεδεμένο δίκτυο

Πως προκύπτει ο αριθμός των παραμέτρων του κάθε επιπέδου;

Ένα πλήρως συνδεδεμένο (πυκνό) στρώμα σε ένα νευρωνικό δίκτυο είναι ένα στρώμα όπου κάθε νευρώνας συνδέεται με νευρώνα στο προηγούμενο στρώμα. Κάθε σύνδεση έχει ένα σχετικό **associated weight** και κάθε νευρώνας έχει μια **bias**.

Έτσι προκύπτει:

Έστω ότι έχω ένα **Dense layer** με:

N μονάδες εισόδου (χαρακτηριστικά)

M μονάδες εισόδου (Νευρώνες σε layer)

Κάθε μονάδα εξόδου σε αυτό το layer θα έχει:

N Weights (αντιστοιχεί ένα για κάθε χαρακτηριστικό εισόδου)

1 bias term

Και έχουμε:

Κάθε ένας από τους **M** νευρώνες έχει **N Weights**.

Επιπλέον, κάθε ένας από τους **M** νευρώνες έχει **1 bias term**.

Έτσι, ο συνολικός αριθμός των παραμέτρων σε αυτό το επίπεδο είναι:

Αριθμός Weights = $N \times M$

Αριθμός biases = M

Έτσι, ο συνολικός αριθμός των παραμέτρων είναι:

Συνολικές παράμετροι = $M \times (N+1)$

Συζήτηση αποτελεσμάτων:

- **Loss: 1.8019**

Η τιμή απώλειας 1,8019 απεικονίζει το μέσο σφάλμα του μοντέλου στην πρόβλεψη των σωστών πιθανοτήτων κλάσης. Χαμηλότερες τιμές απώλειας υποδηλώνουν συνήθως καλύτερη απόδοση του μοντέλου.

Παρόλο που η τιμή απώλειας είναι σχετικά υψηλή, συμβαδίζει με τη χαμηλή ακρίβεια, υποδηλώνοντας ότι οι προβλέψεις του μοντέλου δεν είναι πολύ κοντά στις πραγματικές.

- **Accuracy: 35.15%**

Υποδηλώνει ότι το μοντέλο ταξινόμησε σωστά το 35,15% των εικόνων της δοκιμής. Είναι σχετικά χαμηλό για μια πρακτική εργασία ταξινόμησης εικόνων.

Αυτή η επίδοση υποδηλώνει ότι το πλήρως συνδεδεμένο δίκτυο μπορεί να μην είναι αρκετά πολύπλοκο για να συλλάβει τα περίπλοκα μοτίβα στο σύνολο δεδομένων CIFAR-10.

2.2 Κατηγοριοποίηση με ένα συνελκτικό δίκτυο

Σ' αυτή την περίπτωση, πως προκύπτει ο αριθμός των παραμέτρων;

Απάντηση:

Conv2D Layer 1:

- Input Shape: (32, 32, 3) - Εισόδουμε εικόνες 32x32 pixels με 3 κανάλια (RGB).
- Filters: 16 φίλτρα με διαστάσεις (3, 3).
- Param Calculation: Ο αριθμός των παραμέτρων για κάθε φίλτρο είναι $3 \times 3 \times 3$ (βάση) + 1 (bias) = 28 παραμέτροι.
- Total Param: 16 φίλτρα * 28 παραμέτροι = 448 παραμέτροι.

MaxPooling2D Layer 1:

- Δεν έχει παραμέτρους να εκπαιδεύσει

Conv2D Layer 2:

- Input Shape: (15, 15, 16) - Έξοδος από το πρώτο Conv2D Layer.
- Filters: 32 φίλτρα με διαστάσεις (3, 3).
- Param Calculation: Παρόμοια, ο αριθμός των παραμέτρων για κάθε φίλτρο είναι $3 \times 3 \times 16$ (είσοδος) * 32 + 1 (bias) = 4640 παραμέτροι.

MaxPooling2D Layer 2:

- Δεν έχει παραμέτρους να εκπαιδεύσει

Flatten Layer:

- Δεν έχει παραμέτρους να εκπαιδεύσει

Dense Layer (Output Layer):

- Units: 10 - Ένα πλήρες διασυνδεδεμένο επίπεδο με 10 νευρώνες για τις 10 κατηγορίες του CIFAR-10.
- Param Calculation: Κάθε νευρώνας έχει 1152 εισόδους (από το Flatten Layer) * 10 + 10 (bias) = 11530 παραμέτροι.

Συνολικός Αριθμός Παραμέτρων:

Συνολικές Παράμετροι: 448 (Conv2D Layer 1) + 4640 (Conv2D Layer 2) + 11530 (Dense Layer) = 16618 παράμετροι.

Αυτός ο αριθμός παραμέτρων αντικατοπτρίζει το σύνολο των Weights και των bias που πρέπει να εκπαιδευτούν για την αποδοτική και ακριβή αναγνώριση των κατηγοριών εικόνων στο σύνολο δεδομένων CIFAR-10.

2.3 Πειραματισμός για την ανάπτυξη μοντέλων νευρωνικών δικτύων

Στο πείραμα με τα νευρωνικά δίκτυα, παρατηρώ ότι διάφορες παράμετροι επηρεάζουν σημαντικά τα αποτελέσματα:

Βάθος (Depth) του Δικτύου: Το βάθος του δικτύου, δηλαδή ο αριθμός των κρυφών επιπέδων, έχει σημαντική επίδραση στην ακρίβεια και την επίδοση του μοντέλου. Συγκεκριμένα, όταν αυξάνεται το βάθος, αυξάνεται εκμάθηση πιο σύνθετων χαρακτηριστικών των δεδομένων, κάτι που οδηγεί σε καλύτερη απόδοση του μοντέλου.

Αποτελέσματα πειράματος:

Το μεγαλύτερο βάθος (2) φαίνεται να βελτιώνει την ακρίβεια σε σχέση με το μικρότερο βάθος (1) στις περισσότερες περιπτώσεις. Παραδείγματα είναι τα πειράματα **12** (0.9603) και **15** (0.9754), που έχουν υψηλότερη ακρίβεια από τα αντίστοιχα πειράματα **4** (0.9512) και **7** (0.9734).

Πλάτος (Width) του Δικτύου: Το πλάτος αναφέρεται στον αριθμό των νευρώνων σε κάθε κρυφό επίπεδο. Μεγαλύτερο πλάτος σημαίνει περισσότερες δυνατότητες για παράλληλη επεξεργασία και εκμάθηση των χαρακτηριστικών. Συνήθως, αυτό οδηγεί σε βελτιωμένη απόδοση, αν και μπορεί να αυξήσει τον κίνδυνο υπερεκπαίδευσης, ειδικά όταν τα δεδομένα είναι περιορισμένα.

Αποτελέσματα πειράματος:

Το μεγαλύτερο πλάτος (64 νευρώνες) φαίνεται να προσφέρει καλύτερη απόδοση σε σύγκριση με το μικρότερο πλάτος (32 νευρώνες). Αυτό φαίνεται ξεκάθαρα στα πειράματα **7** και **15**, όπου το πλάτος 64 νευρώνων οδηγεί σε υψηλότερη ακρίβεια (0.9734 και 0.9754) σε σύγκριση με τα πειράματα **3** και **11** (0.9678 και 0.9664).

Ρυθμός Εκμάθησης (Learning Rate): Ο ρυθμός εκμάθησης καθορίζει πόσο γρήγορα το μοντέλο μαθαίνει από τα δεδομένα. Ένας πολύ μικρός ρυθμός εκμάθησης μπορεί να οδηγήσει σε πιο αργή σύγκλιση του μοντέλου, αλλά μπορεί να βοηθήσει στην επίτευξη καλύτερης γενίκευσης και απόδοσης. Αντίθετα, ένας πολύ μεγάλος ρυθμός εκμάθησης μπορεί να οδηγήσει σε ασταθή εκπαίδευση και υπερεκπαίδευση.

Αποτελέσματα πειράματος:

Ένας μικρότερος ρυθμός εκμάθησης (0.001) προσφέρει καλύτερα αποτελέσματα σε σχέση με έναν μεγαλύτερο ρυθμό (0.01). Παραδείγματα είναι τα πειράματα **7** και **15** με ρυθμό εκμάθησης 0.001, που έχουν ακρίβεια 0.9734 και 0.9754 αντίστοιχα, σε σύγκριση με τα πειράματα **5** και **13** με ρυθμό εκμάθησης 0.01, που έχουν ακρίβεια 0.9641 και 0.9672 αντίστοιχα.

Χρήση Optimizer: Οι διαφορετικοί optimizers (όπως ο SGD και ο Adam) έχουν διαφορετικούς τρόπους να ενημερώνουν τις παραμέτρους του μοντέλου κατά τη διάρκεια της εκπαίδευσης. Ο Adam συνήθως προσφέρει καλύτερη απόδοση λόγω της ικανότητάς του να αντιμετωπίζει τα προβλήματα της κλίσης κατά την εκπαίδευση.

Αποτελέσματα πειράματος:

Ο Adam optimizer δείχνει να αποδίδει καλύτερα από τον SGD στις περισσότερες περιπτώσεις. Παραδείγματα είναι τα πειράματα **7** (0.9734) και **15** (0.9754) με Adam optimizer, που έχουν υψηλότερη ακρίβεια σε σύγκριση με τα πειράματα **6** (0.9013) και **14** (0.9074) με SGD optimizer.

Exp	Depth	Width	Optimizer	Learning Rate	Loss	Accuracy
0	1	32	SGD	0.010	0.190752	0.9450
1	1	32	Adam	0.010	0.185556	0.9598
2	1	32	SGD	0.001	0.383145	0.8946
3	1	32	Adam	0.001	0.113651	0.9678
4	1	64	SGD	0.010	0.167801	0.9512
5	1	64	Adam	0.010	0.200101	0.9641
6	1	64	SGD	0.001	0.364352	0.9013
7	1	64	Adam	0.001	0.093503	0.9734
8	2	32	SGD	0.010	0.155078	0.9558
9	2	32	Adam	0.010	0.190540	0.9558
10	2	32	SGD	0.001	0.343654	0.9014
11	2	32	Adam	0.001	0.112906	0.9664
12	2	64	SGD	0.010	0.134611	0.9603
13	2	64	Adam	0.010	0.162980	0.9672
14	2	64	SGD	0.001	0.328288	0.9074
15	2	64	Adam	0.001	0.093455	0.9754

Καθώς αυξάνεται η συνθετότητα ενός μοντέλου, π.χ. αυξάνοντας το βάθος και το πλάτος του, οι επιδόσεις του μπορούν να βελτιωθούν εάν αυτά τα βήματα γίνονται σωστά. Ωστόσο, η προσθήκη περισσότερων επιπέδων και νευρώνων αυξάνει την πολυπλοκότητα του μοντέλου και τον κίνδυνο υπερεκπαίδευσης, εάν δεν ελέγχονται σωστά άλλοι παράγοντες όπως η κανονικοποίηση.

3. PART 3

3.1 Πειραματισμός για την ανάπτυξη μοντέλων νευρωνικών δικτύων

Ποιες παράμετροι φαίνεται να επηρεάζουν τα αποτελέσματα; Τι παρατηρείτε όσο αυξάνεται η συνθετότητα ενός μοντέλου;

Αποτελέσματα Μοντέλων:

- **Γραμμική Παλινδρόμηση**

Mean Squared Error (MSE): 0.7089985923456071

R-squared (R^2): 0.5249824155975751

- **Random Forest:**

Random Forest - Mean Squared Error (MSE): 0.7736736263736262

Random Forest - R-squared (R^2): 0.4816511893203884

Συγκρίνοντας τα αποτελέσματα των δύο μοντέλων:

- **Mean Squared Error (MSE):** Το μοντέλο γραμμικής παλινδρόμησης έχει μικρότερο MSE (0.709 εναντίον 0.774 για το τυχαίο δάσος), προσδιορίζοντας τη γραμμική παλινδρόμηση ως καλύτερη επιλογή για τη μείωση του τετραγωνικού μέσου σφάλματος.
- **R-squared (R^2):** Το R^2 για τη γραμμική παλινδρόμηση είναι επίσης υψηλότερο (0.525 εναντίον 0.482 για το τυχαίο δάσος), υποδεικνύοντας ότι η γραμμική παλινδρόμηση εξηγεί καλύτερα τη μεταβλητότητα του στόχου σε σχέση με το τυχαίο δάσος.

Συμπέρασμα:

Βάσει αποτελεσμάτων, η γραμμική παλινδρόμηση εμφανίζεται ως καλύτερη επιλογή από το τυχαίο δάσος για την πρόβλεψη. Παρά το γεγονός ότι το τυχαίο δάσος είναι ένα πιο πολύπλοκο μοντέλο που μπορεί να προσαρμοστεί καλύτερα στα δεδομένα εκπαίδευσης, φαίνεται ότι η γραμμική παλινδρόμηση παρέχει καλύτερη γενίκευση στα νέα δεδομένα ελέγχου.

Παράμετροι που επηρεάζουν τα αποτελέσματα:

- **Depth:** η αύξηση των επιπέδων μπορεί να βελτιώσει την ικανότητα του μοντέλου να αναγνωρίζει πιο πολύπλοκα μοτίβα στα δεδομένα, αλλά μπορεί επίσης να οδηγήσει σε υπερεκπαίδευση (overfitting) αν δεν υπάρχουν αρκετά δεδομένα ή αν δεν χρησιμοποιηθούν σωστά οι τεχνικές κανονικοποίησης.
- **Width:** Ο αριθμός των νευρώνων σε κάθε επίπεδο επηρεάζει επίσης την ικανότητα του μοντέλου να γενικεύει τα δεδομένα. Μεγαλύτερο πλάτος μπορεί να προσφέρει μεγαλύτερη ακρίβεια, αλλά αυξάνει επίσης την πολυπλοκότητα του μοντέλου.