



ΧΑΡΟΚΟΠΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ
HAROKOPIO UNIVERSITY

Εργασία 2

Γραμμικά μοντέλα στο scikit-learn

Τεχνητή Νοημοσύνη

Νικόλαος Γουρνάκης, it22023

13 Ιουνίου 2024

Ερώτημα 1)

1)

Για να απαντήσω στο ποσες εγγραφές έχουμε στο σύνολο δεδομένων χρησιμοποιώ το shape attribute του dataframe

```
print(data.shape)
```

```
(12330, 18)
```

Όπου το πρώτο νούμερο είναι το σύνολο εγγραφών και το δεύτερο είναι το σύνολο στηλών/features

Αρα

```
print(data.shape[0])
```

εχουμε 12330 εγγραφές.

2)

ποσοστό από αυτές οι χρήστες αγόρασαν τελικά = $\text{σύνολο ατομών που αγόρασαν} / \text{σύνολο ατομών} * 100$. Τα άτομα που αγόρασαν είναι αυτά που στο 'Revenue' είναι True

```
print(data[data['Revenue'] == True].shape[0] / data.shape[0] * 100)
```

```
15.474452554744525
```

αρα το ποσοστό από αυτές οι χρήστες αγόρασαν τελικά είναι περίπου 15.47%

3)

Για αυτό το σενάριο ξέρουμε ότι το μοντέλο πάντα προβλέπει False αρα για να υπολογίσουμε την ευστοχία πρέπει να κάνουμε = $\text{σύνολο ατομών που ΔΕΝ αγόρασαν} / \text{σύνολο ατομών} * 100$. Μια εναλλακτική θα ήταν να κάνουμε αφαιρέσουμε το αποτέλεσμα του προηγούμενου ερωτήματος από το 100 και θα έχουμε την ίδια απάντηση.

```
print(data[data['Revenue'] == False].shape[0] / data.shape[0] * 100)
```

```
84.52554744525548
```

Το accuracy ενός μοντέλου που πάντα επιστρέφει False είναι περίπου 84.52%

Ερώτημα 5)

Ευστοχία μοντελου στο συνολο εκπαιδευσης:

```
print(model.score(X_train, y_train))
```

```
0.8767234387672344
```

Ευστοχία μοντελου στο συνολο δοκιμης:

```
print(model.score(X_test, y_test))
```

```
0.8745606920789403
```

Πινακας συγχυσης:

```
print(confusion_matrix(y_test, model.predict(X_test)))
```

```
[[3083  41]
 [ 423 152]]
```

Ερμηνεια πινακα συγχυσης:

Αμα δουμε τα docs της συναρτησης confusion_matrix

Returns: Confusion matrix whose i-th row and j-th column entry indicates the number of samples with true label being i-th class and predicted label being j-th class.

Βλεπουμε οτι η γραμμες ειναι true classes και οι στηλες predicted classes αρα αυτο που μας λεει ο πινακας συγχυσης ειναι το ακολουθο

	Το μοντελο πιστευει οτι ΔΕΝ αγορασε	Το μοντελο πιστευει οτι αγορασε
Οντως ΔΕΝ αγορασε	[[3083	41]
Οντως αγορασε	[423	152]]

Αρα 3083 άτομα στο τεστ το μοντελο καταφερε να μαντέψει σωστα οτι δεν θα αγοράσουν και 423 λαθος οτι δεν θα αγοράσουν ενω στην πραγματικοτητα θα αγοράσουν. Και 152 άτομα στο τεστ το μοντελο καταφερε να μαντέψει σωστα οτι θα αγοράσουν και 41 λαθος οτι θα αγοράσουν ενω πραγματικα δεν θα αγοράσουν. Το accuracy του μοντελου βγαινει απο την διαγωνιο / το συνολο δηλαδή $= (3083 + 152) / (3083 + 152 + 423 + 41) = 3235 / 3699 \approx 0.87$. Βλεπω οτι το μοντελο δινει παραπανω βαρος στο να μαντέψω κατι σαν “μη αγορά” καθώς εχουμε 423 False Negatives, αυτο λογικα γινεται γιατι η κατανομη των κλασεων ειναι παρα πολυ ανησωροπη.

Τροποποιησεις:

Το κυριο προβλημα ειναι αυτη η ανισορροπια μεταξυ των δυο κλασεων, ξερουμε οτι στο συνολο δεδομενων [τα άτομα που θα αγοράσουν ειναι 15.47%](#) αρα η True class ειναι 15.47% και η False class ειναι το υπολοιπο 84.53% που ειναι τεραστια διαφορα.

Αυτο μπορει να προκαλεσει προβληματα:

- Το μοντελο δινει πιο πολυ βαρος στο False class καθώς ειναι πιο πιθανο να ειναι σωστο
- Με το μικρο ποσοστο του True class εχουμε υψηλη πιθανοτητα να γινει καταστροφικος διαχωρισμος των δεδομενων, δηλαδή ολα τα False class να πεσουν στο training set και ολα τα True class να πεσουν στο Testing set, που αυτο εχει ως αποτελεσμα το μοντελο να μαντεύει παντα false.

Αρα τροποι βελτιωσεις:

- Καταρχας θα ηταν καλο να κανουμε KFold validation για να σιγουρευτούμε οτι δεν κανουμε overfit.
- Θα μπορούσαμε να βαλουμε βαρος στην καθε κλαση ετσι ωστε το μοντελο να κρινει πιο σημαντικο να πετυχει σωστα την κλαση μειονοτητα.
- Θα μπορούσαμε να κανουμε stratification στο train_test_split ετσι ωστε να υπαρχουν αρκετα δεδομενα και απο τις δυο κλασεις και στα δυο σετ.
- Θα μπορούσαμε να κανουμε undersampling ετσι ωστε να φερουμε το νουμερο το κλασεων στο 50/50
- Θα μπορούσαμε να κανουμε [SMOTE](#) για να αυξήσουμε συνθετικα το νούμερο της κλασης μειονοτητα