Ανάλυση Δεδομένων

Υπεύθυνη καθηγήτρια: κα. Παναγιωτίδου Σοφία

4η εργασία

Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης
Τμήμα Μηχανολόγων Μηχανικών
31/05/2024

Δημήτριος Νεντίδης, 6821

Περιεχόμενα

1 Εισαγωγή

Αυτή η εργασία αποτελεί στην ουσία συνέχεια της προηγούμενης. Ενώ στην 1η γινόταν σύγκριση των μοντέλων βάση παραμέτρων όπως το R^2 ή το p-value για να εξαχθούν συμπεράσματα σχετικά με το fit ή την στατιστική σημαντικότητα αντίστοιχα, εδώ, στο πρώτο μέρος γίνεται σύγκριση μεθόδων με κριτήριο το μέσο σφάλμα πρόβλεψης των testing data (test MSE) που προκύπτει από τα διαφορετικά runs της μεθόδου διασταυρωμένης επικύρωσης. Οι μέθοδοι που συγκρίνονται είναι η πολλαπλή παλινδρόμηση που εξετάστηκε στο 2ο υποερώτημα της 1ης εργασίας και η forward selection που εξετάστηκε στο 3ο. Στην συνέχεια πραγματοποιήται μια επιπλέον σύγκριση της k-fold cross-validation με την Leave-One-Out Cross-Validation για την καλύτερη εκ των δύο μεθόδων που εξετάστηκαν με την πρώτη.

Στο δεύτερο μέρος, χωρίζονται τα δεδομένα σε δύο υποσύνολα, το training και το testing set και μετατρέπεται η απόκριση σε δυαδική μεταβλητή, ένα για όποια τιμή της είναι πάνω από το μέσο όρο και 0 όταν είναι κάτω. Στην συνέχεια τρέχουν κάποιες αναλύσεις με διαφορετικές παραμέτρους, εξετάζοντας κάθε φορά την εκτίμηση της ακρίβειας στα testing αλλά και στα training data, εξάγωντας συμπεράσματα για το bias και το variance των μοντέλων. Οι μέθοδοι που εξετάζονται εδώ είναι η μέθοδος της Λογιστικής Παλινδρόμησης, η Linear Discriminant Analysis καθώς και η μέθοδος K-nearest-neighbors.

2 Μέρος Α

Στο πρώτο μέρος, είναι σημαντικό να παρατηρηθεί ότι δεν γίνεται σύγκριση μοντέλων αλλά μεθόδων. Δεν υπάρχει κάποιο συγκεκριμένο μοντέλο που εξετάζεται προκύπτουν παραπάνω από ένα μοντέλα για κάθε μέθοδο και λαμβάνεται υπ' όψη ο μέσος όρος της επίδοσης τους, ο οποίος θεωρείται αντιπροσωπευτικός για κάθε μέθοδο.

2.1 Μέθοδος διασταυρούμενης επικύρωσης

Στην περίπτωση της μεθόδου διασταυρούμενης επικύρωσης, με k=5, προκύπτουν συνολικά 4 μοντέλα. Εδώ, το σύνολο των δεδομένων χωρίζεται σε 5 υποσύνολα, 4 εκ των οποίων σχηματίζουν το σετ των

training data και το 5ο λαμβάνει τον ρόλο του testing set. Η διαδικασία αυτή επαναλαμβάνεται μέχρι να περάσουν όλα τα υποσύνολα από τον ρόλο του testing set.

Σε κάθε run, τα training data χρησιμοποιούνται για να βρεθεί το βέλτιστο μοντέλο με το καλύτερο fit, μετρώντας στην πορεία το training MSE. Εν συνεχεία, τα εναπομείναντα δεδομένα του 5ου σετ χρησιμοποιούνται για να αξιολογηθεί το μοντέλο που προέκυψε, βγάζοντας έτσι το test MSE.

Στο παρόν υποερώτημα το ζητούμενο είναι να αξιολογηθούν διαφορετικές μέθοδοι για την εύρεση του βέλτιστου μοντέλου, το μοντέλο με το ελάχιστο training MSE. Οι δύο μέθοδοι που εξετάζονται είναι **Πολλαπλή Γραμμική Παλινδρόμηση**, όπου λαμβάνονται υπόψη όλες οι προβλεπτικές μεταβλητές και η **Forward Selection**, όπου επιλέγονται μεταβλητές που περνάνε ένα φράγμα σημαντικότητας και βελτιώνουν το μοντέλο. Στην προκειμένη περίπτωση, ορίζοντας το p-value, επίπεδο σημαντικότητας, σε διαφορετικές τιμές αλλάζει το ποια από τις δύο μεθόδους έχει το πάνω χέρι από πλευρά επίδοσης σε σχέση με το test MSE. Τα αποτελέσματα συνοψίζονται στον παρακάτω πίνακα:

p-value	**Forward Selection MSE**
0.05	13.3707
0.01	13.3925
0.001	13.5223

Πίνακας 1: Μέσο test MSE για διαφορετικά p-value.

Η Πολλαπλή Γραμμική Παλινδρόμηση έχει, ανεξάρτητα από τα 3 πιθανά p-values που εξετάστηκαν το ίδιο μέσο εκτιμώμενο σφάλμα στα test data, MSE = 13.3794. Είναι εμφανές ότι όσο μειώνεται το p-value τόσο υπερισχύει η μέθοδος της Πολλαπλής Γραμμικής Παλινδρόμησης. Ωστόσο, το ελάχιστο μέσο test MSE συνολικά εμφανίζεται για p-value ίσο με 0.05 και την μέθοδο Forward selection. Συνεπώς με αυτές τις παραμέτρους θα προχωρήσει η ανάλυση στο επόμενο ερώτημα.

2.2 Leave-One-Out Cross-Validation

Αυτή η μέθοδος είναι ουσιαστικά υποπερίπτωση της μεθόδου διασταυρούμενης επικύρωσης. Αντί να χωρίστουν τα δεδομένα σε k υπο-

σύνολα, κάθε παρατήρηση αποτελεί από μόνη της ένα σετ. Όπως περιγράφεται και στο όνομα της μεθόδου, κάθε φορά το training υποσύνολο αποτελείται από όλες τις παρατηρήσεις πλην μιας, η οποία αποτελεί το testing set. Όπως και πριν, η διαδικασία συνεχίζει μέχρι να περάσουν όλες οι παρατηρήσεις από τον ρόλο του testing set. Είναι εύκολα κατανοητό ότι το k εδώ είναι ίσο με το πλήθος των παρατηρήσεων.

Σε αυτό το υποερώτημα πραγματοποιήται μια σύγκριση της επικρατέστερης μεθόδου που προέκυψε από το προηγούμενο ερώτημα με την k-fold-cross-validation, με την επίδοση της μεθόδου αυτής με την χρήση της LOOCV. Στην προκειμένη περίπτωση, επιλέγεται p-value ίσο με 0.05 και η μέθοδος που προκύπτει είναι η Forward Selection με test MSE ίσο με 13.3707. Τρέχοντας τον αλγόριθμο προκύπτει ότι με την χρήση της Leave-One-Out και επίπεδο σημαντικότητας 0.05 για την επιλογή μεταβλητών με πρόσω επιλογή, το test MSE πέφτει στο 13.2232, κάτω από την τιμή που είχε προκύψει με μόνο 5 υποσύνολα.

Η μέθοδοσ LOOCV χρησιμοποιεί μεγαλύτερο μέρος των δεδομένων κάθε φορά, έχοντας όλα τα δεδομένα πλην μιας παρατήρησης, μειώνοντας έτσι δραστικά το bias. Ωστόσο, το ίδιο γεγονός μπορεί να οδηγήσει σε αυξημένο variance για συγκεκριμένα μοντέλα που επηρεάζονται από outliers. Από την άλλη έχοντας μόνο 5 σετ στην 5-fold, μπορεί να δημιουργήσει μοτίβα τα οποία επηρεάζουν την επίδοση και ρίχνουν την συνολική επίδοση της μεθόδου. Στην προκειμένη περίπτωση, τόσο το variance της 5-fold είναι μεγαλύτερο από αυτό της LOOCV όσο και το bias είναι μεγαλύτερο, μη έχοντας την πρόσβαση στα ίδια δεδομένα. Συνεπώς το test MSE της LOOV θεωρείται πιο αξιόπιστο.

Τέλος αξιοσημείωτο είναι ότι συνολικά ο χρόνος που χρειάστηκε για την εφαρμογή αυτής της μεθόδου είναι τάξεις μεγέθους περισσότερος από ότι για την k-fold-cross-validation. Αυτό είναι απολύτως λογικό κρίνοντας από το πόσες φορές χρειάστηκε να επαναληφθεί η διαδικασία με διαφορετικά σετ. Ίσως μια καλή μέση λύση θα ήταν η αύξηση των folds από 5 σε 10, ο χρόνος δεν θα αυξανόταν τόσο δραστικά και θα υπήρχε πιο αξιόπιστη εκτίμηση του μέσου σφάλματος.

3 Μέρος Β

Όπως αναφέρθηκε και στην εισαγωγή, σε αυτό το μέρος μετατρέπεται η απόκριση, Life Expectancy, σε δυαδική μεταβλητή, λαμβάνοντας την τιμή 0 όταν η απόκριση είναι κάτω από το μέση τιμή της και 1 όταν είναι πάνω. Έπειτα χωρίζονται τα δεδομένα σε training data, 80% του συνόλου των παρατηρήσεων και σε testing data 20% των παρατηρήσεων. Περαιτέρω πληροφορίες για τα μοντέλα και τους συντελεστές των προβλεπτικών μεταβλητών μπορούν να βρεθούν στην αντίστοιχη ενότητα, δεν συμπεριλαμβάνονται σε αυτό το μέρος.

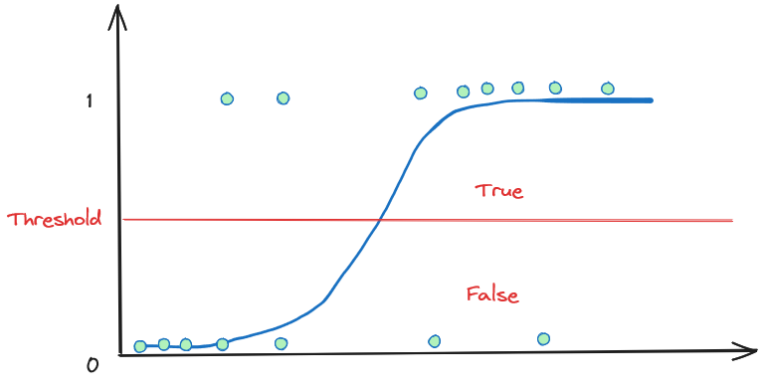
Τα γενικά πλαίσια που γίνεται η σύγκριση των μοντέλων είναι η σχέση bias και variance. Το bias είναι ένα μέτρο προσδιορισμού του πόσο καλό είναι το fit του μοντέλου στα training data, ενώ το variance περιγράφει στην ουσία το πόσο καλά το μοντέλο μπορεί να γενικευθεί σε δεδομένα που δεν έχει ξαναδεί.

Για να συγκριθούν τα μοντέλα χρησιμοποιούνται οι εξής παράμετροι. Πρώτα πρώτα τα testing και training errors. Ο τρόπος που ορίζονται αυτά είναι υπολογίζοντας πρώτα την ακρίβεια κάθε μοντέλου και έπειτα αφαιρώντας την ακρίβεια αυτή από την μονάδα. Όσο μικρότερο το error τόσο μεγαλύτερη ακρίβεια έχει το μοντέλο στο fit των αντίστοιχν δεδομένων, είτε πρόκειτε για training είτε για testing. Παραδείγματος χάρη για την λογιστική παλινδρόμηση είναι το ακόλουθο κομμάτι κώδικα.

```
# Calculate accuracies and errors
train_accuracy_lr = accuracy_score(y_train, y_train_pred_lr)
test_accuracy_lr = accuracy_score(y_test, y_test_pred_lr)
train_error_lr = 1 - train_accuracy_lr
test_error_lr = 1 - test_accuracy_lr
```

Έπειτα, είναι σημαντικό να ληφθούν υπόψη οι πίνακες σύγχυσης. Στην ουσία είναι πίνακες που δείχνουν πόσες παρατηρήσεις προβλέφθηκαν σωστά (true positives, 1, και true negatives, 4) και πόσες λάθος (false positives, 3, και false negatives, 2). Οι πίνακες αυτοί αποτελούν ένα ιδιαίτερα χρήσιμο εργαλείο καθώς πέρα από την κατανόηση της φύσης του στατιστικού λάθους μας επιτρέπουν να έχουμε μια ιδιαίτερα σαφή εικόνα της κατάταξης των παρατηρήσεων.

Την εικόνα συμπληρώνει το classification report. Στην πρώτη στήλη του δίνει ποσοστιαία την επιτυχή κατανομή των παρατηρήσεων στον



Σχήμα 1: Σχηματικό threshold.

πίνακα σύγχυσης, το recall που ορίζεται ως ο αριθμός των true positives διαιρεμένος με το άθροισμα των true positives false negatives, δηλαδή στην ουσία το πλήθος όλων των positives, το f1-score που είναι ο μέσος όρος των precision και recall.

Πριν προχωρήσουμε στις μεθόδους είναι κρίσιμο να δωθεί μια εξήγηση και για την φύση του threshold. Στην ουσία, αποτελεί την τιμή μεταξύ 0 και 1 πάνω από την οποία μια απόκριση θεωρείται high.

3.1 Λογιστική Παλινδρόμηση

Η Λογιστική Παλινδρόμηση αποτελεί μια μέθοδο για την κατάταξη των αποκρίσεων των παρατηρήσεων σε δύο επίπεδα τιμών, ψηλό και χαμηλό. Στην πραγματικότητα η Λογιστική Παλινδρόμηση, σαν διαδικασία, μοιάζει πάρα πολύ με την Πολλαπλή Γραμμική Παλινδρόμηση, με διαφορά ότι εδώ οι συντελεστές που προκύπτουν για τις μεταβλητές συσχετίζουν τις παρατηρήσεις με τον φυσικό λογάριθμο της πιθανότητας να έχουμε high σε μια παρατήρηση, κάνοντας έναν μετασχηματισμό στα δεδομένα.

Τα μοντέλα που προέκυψαν είναι τα παρακάτω:

3.1.1 Threshold = 0.4

Training Data:

```
Confusion Matrix:
[[453, 134],
 [ 51, 681]]
```

Classification Report:

	precision	recall	f1-score	support
0	0.90	0.77	0.83	587
1	0.84	0.93	0.88	732

Testing Data:

```
Confusion Matrix:
[[ 97,  30],
 [ 14, 189]]
```

Classification Report:

	precision	recall	f1-score	support
0	0.87	0.76	0.82	127
1	0.86	0.93	0.90	203

3.1.2 Threshold = 0.5

Training Data:

```
Confusion Matrix:
[[490,  97],
 [ 84, 648]]
```

Classification Report:

	precision	recall	f1-score	support
0	0.85	0.83	0.84	587
1	0.87	0.89	0.88	732

Testing Data:

```
Confusion Matrix:
[[103,  24],
 [ 25, 178]]
```

Classification Report:

	precision	recall	f1-score	support
0	0.80	0.81	0.81	127
1	0.88	0.88	0.88	203

3.1.3 Threshold = 0.6

Training Data:

```
Confusion Matrix:
[[512,  75],
 [129, 603]]
```

Classification Report:

	precision	recall	f1-score	support
0	0.80	0.87	0.83	587
1	0.89	0.82	0.86	732

Testing Data:

```
Confusion Matrix:
[[109,  18],
 [ 40, 163]]
```

Classification Report:

	precision	recall	f1-score	support
0	0.73	0.86	0.79	127
1	0.90	0.80	0.85	203

Βλέποντας τα δεδομένα παραπάνω είναι εμφανές ότι η μεγαλύτερη ακρίβεια παρουσιάζεται για threshold = 0.5 όσον αφορά το training error και για 0.4 όσον αφορά το testing error. Ακολουθεί μια σύνοψη στον ακόλουθο πίνακα.

Συνολικά όλα τα μοντέλα δείχνουν να έχουν σχετικά χαμηλό bias και παρόμοιο error στα δύο σετ δεδομένων το οποίο δείχνει και χαμηλό

Threshold	Training Error	Testing Error
0.4	0.1403	0.1333
0.5	0.1372	0.1485
0.6	0.1547	0.1758

Πίνακας 2: Λογιστική Παλινδρόμηση σφάλματα.

variance. Για threshold 0.6 αυξάνονται και τα δύο αυτά μεγέθη. Η αλλαγή του threshold φαίνεται να επηρεάζει το specificity (true negative rate) και το sensitivity (true positive rate) των μοντέλων. Μείωση του threshold οδηγεί σε αύξηση του specificity, ενώ αύξηση του οδηγεί σε αύξηση του sensitivity. Το γεγονός ότι το testing error είναι μικρότερο από το training error για 0.4, παρότι ασυνήθιστο, ενδέχεται να είναι εξαιτίας συγκεκριμένων τιμών που κάνουν καλύτερο fit για το συγκεκριμένο μοντέλο.

Αξίζει να σημειωθεί ότι, με την εφαρμογή του πακέτου sci-kit learning, για αυτή την μέθοδο δεν αρκούσαν οι επαναλήψεις που έτρεχε ο αλγόριθμος για την βελτιστοποίηση του μοντέλου. Συνολικά αυξήθηκε ο αριθμός των επαναλήψεων σε 5000, χωρίς όμως να λυθεί το πρόβλημα, παρά την αύξηση του υπολογιστικού χρόνου.

3.2 Linear Discriminant Analysis

Η LDA προσπαθεί να ταξινομήσει τα δεδομένα βρίσκοντας γραμμικές συσχετίσεις μεταξύ των προβλεπτικών μεταβλητών με στόχο να μειώσει τις διαστάσεις του μοντέλου.

3.2.1 Threshold = 0.4

Training Data

```
Confusion Matrix:
 [[458, 129],
  [ 22, 710]]
Classification Report:
              precision    recall  f1-score   support
           0       0.95      0.78      0.86       587
```

```
           1       0.85      0.97      0.90       732
```

Testing Data

```
Confusion Matrix:
 [[101, 26],
  [  3, 200]]
Classification Report:
              precision    recall  f1-score   support
           0       0.97      0.80      0.87       127
           1       0.88      0.99      0.93       203
```

3.2.2 Threshold = 0.5

Training Data

```
Confusion Matrix:
 [[481, 106],
  [ 40, 692]]
Classification Report:
              precision    recall  f1-score   support
           0       0.92      0.82      0.87       587
           1       0.87      0.95      0.90       732
```

Testing Data

```
Confusion Matrix:
 [[105, 22],
  [  8, 195]]
Classification Report:
              precision    recall  f1-score   support
           0       0.93      0.83      0.88       127
           1       0.90      0.96      0.93       203
```

3.2.3 Threshold = 0.6

Training Data

```
Confusion Matrix:
 [[496, 91],
  [ 62, 670]]
Classification Report:
              precision    recall  f1-score   support
           0       0.89      0.84      0.87       587
           1       0.88      0.92      0.90       732
```

Testing Data

```
Confusion Matrix:
 [[107, 20],
  [ 13, 190]]
Classification Report:
              precision    recall  f1-score   support
           0       0.89      0.84      0.87       127
           1       0.90      0.94      0.92       203
```

Threshold	Training Error	Testing Error
0.4	0.1145	0.0879
0.5	0.1107	0.0909
0.6	0.1160	0.1000

Πίνακας 3: LDA σφάλματα.

Σε γενικές γραμμές τόσο το bias όσο και το variance στα μοντέλα της LDA είναι χαμηλά. Τα μοντέλα έχουν ένα αρκετά ικανοποιητικό fit των training data, ενώ είναι ξεκάθαρο ότι γενικεύουν πολύ καλά, τα δύο σφάλματα είνια πολυ κοντά. Αλλαγές στο threshold δεν επηρεάζουν την επίδοση του μοντέλου δραστικά και επιτρέπουν την δυνατότητα fine tuning για βελτίωση του sensitivity και του specificity χωρίς να επηρεάζεται δραστικά το ένα από το άλλο, όπως συνέβαινε στην logistic regression. Το specificity φαίνεται να είναι αυξημένο σχεδόν κάθε φορά, ειδικά με 0.4, χωρίς να μικραίνει το sensitivity.

Ενδιαφέρον παρουσιάζει το γεγονός ότι παρά τα αρκετά μικρότερα σφάλματα που προκύπτουν με την χρήση αυτής της μεθόδου, το test error είναι μικρότερο από το training error για κάθε τιμή του threshold. Μια πιθανή εξήγηση είναι ότι οι παρατηρήσεις του training set έχουν κάποια Outliers τα οποία αυξάνουν το training error. Κάτι τέτοιο δεν θεωρείται πιθανό καθώς όλες οι μέθοδοι αξιολογούνται με την χρήση των ίδιων δεδομένων. Αυξάνοντας το μέγεθος του training set από 80% του συνόλου των δεδομένων σε 90%, και για threshold ίσο με 0.5, παρατηρείται ότι το σφάλμα των training data πέφτει κάτω από το σφάλμα των testing data, με τιμές 0.1037 και 0.1151 αντίστοιχα, ωστόσο αυτό δεν αποτελεί μέρος της παρούσας αναφοράς και συνεπώς δεν θα γίνει εκτενής αναφορά στο μοντέλο.

3.3 K-nearest-neighbors

Η μέθοδος K-nearest-neighbors βασίζεται στην ιδέα ότι σε ένα διάγραμμα κάποιων παραμέτρων διάστασης N, οι γειτονικές παρατηρήσεις θα έχουν κοινή απόκριση. Συνεπώς μπορεί να προβλεφθεί η απόκριση λαμβάνοντας υπόψη την τοποθεσία μιας παρατήρησης στον χώρο των παρατηρήσεων. δημιουργείται ένας υποχώρος N διαστάσεων που περιέχει μέσα K γειτονικές παρατηρήσεις. Στην συνέχεια, κατατάσεται η απόκριση βάση της απόκρισης της πλειοψηφίας των K κοντινότερων γειτώνων.

3.3.1 Neighbors k = 1

Training Data

```
Confusion Matrix:
 [[587   0]
 [  0 732]]
Classification Report:
              precision    recall  f1-score   support
           0       1.00      1.00      1.00       587
           1       1.00      1.00      1.00       732
```

Testing Data

```
Confusion Matrix:
 [[ 77  50]
 [ 70 133]]
Classification Report:
              precision    recall  f1-score   support
           0       0.52      0.61      0.56       127
           1       0.73      0.66      0.69       203
```

3.3.2 Neighbors k = 3

Training Data

```
Confusion Matrix:
 [[477 110]
 [140 592]]
Classification Report:
              precision    recall  f1-score   support
           0       0.77      0.81      0.79       587
           1       0.84      0.81      0.83       732
```

Testing Data

```
Confusion Matrix:
 [[ 81  46]
 [ 69 134]]
Classification Report:
              precision    recall  f1-score   support
           0       0.54      0.64      0.58       127
           1       0.74      0.66      0.70       203
```

3.3.3 Neighbors k = 5

Training Data

```
Confusion Matrix:
 [[432 155]
 [155 577]]
```

```
Classification Report:
              precision    recall  f1-score   support
           0       0.74      0.74      0.74       587
           1       0.79      0.79      0.79       732
```

Testing Data

```
Confusion Matrix:
 [[ 86  41]
 [ 75 128]]
Classification Report:
              precision    recall  f1-score   support
           0       0.53      0.68      0.60       127
           1       0.76      0.63      0.69       203
```

3.3.4 Neighbors k = 7

Training Data

```
Confusion Matrix:
 [[421 166]
 [188 544]]
Classification Report:
              precision    recall  f1-score   support
           0       0.69      0.72      0.70       587
           1       0.77      0.74      0.75       732
```

Testing Data

```
Confusion Matrix:
 [[ 84  43]
 [ 75 128]]
Classification Report:
              precision    recall  f1-score   support
           0       0.53      0.66      0.59       127
           1       0.75      0.63      0.68       203
```

3.3.5 Neighbors k = 9

Training Data

```
Confusion Matrix:
 [[389 198]
 [187 545]]
Classification Report:
              precision    recall  f1-score   support
           0       0.68      0.66      0.67       587
           1       0.73      0.74      0.74       732
```

Testing Data

```
Confusion Matrix:
 [[ 81  46]
 [ 71 132]]
Classification Report:
              precision    recall  f1-score   support
           0       0.53      0.64      0.58       127
           1       0.74      0.65      0.69       203
```

Number of Neighbors	Error	Testing Error
1	0.0000	0.3636
3	0.1895	0.3485
5	0.2350	0.3515
7	0.2684	0.3576
9	0.2919	0.3545

Πίνακας 4: KNN σφάλματα.

Όπως είναι λογικό με μόνο ένα γείτονα το training error δεν μπορεί παρά είναι μηδενικό. Στην ουσία αυτό μας λέει απλά ότι ο γείτονας πάντα είναι ο γείτονας. Όσο ανεβαίνει ο αριθμός των γειτόνων αυξάνεται και το training error, πλησιάζοντας το testing error, το οποίο δεν αλλάζει δραστικά. Για έναν γείτονα είναι το μεγαλύτερο, ωστόσο η διαφορά είναι ελαφρώς παραπάνω από 4% από την καλύτερη επίδοση,

τους δύο γείτονες. Αυτό που κάνει εντύπωση είναι ότι ανεξάρτητα από τον αριθμό των γειτόνων, η ακρίβεια είναι πολύ καλύτερη στην πρόβλεψη των θετικών σε σχέση με των αρνητικών. Άρα εν ολίγεις η μέθοδος αυτή έχει χαμηλό bias και υψηλό variance για το δεδομένο σετ παρατηρήσεων.

Η μέθοδος αυτή διαφέρει από μεθόδους όπως η Λογιστική Παλινδρόμηση. Εδώ δεν υπάρχουν συντελεστές για τις προβλεπτικές μεταβλητές, ή intercept για την κλίση.

3.4 Σύγκριση

Συνοψίζοντας όλα τα παραπάνω, μπορεί να ειπωθεί ότι η LDA έχει σταθερά training και testing errors λιγότερο από την λογιστική παλινδρόμηση, έχοντας συνολικά καλύτερο fit και καλύτερη ακρίβεια. Παράλληλα έχει χαμηλότερο variance έχοντας καλύτερη ικανότητα να γενικεύονται τα μοντέλα της σε δεδομένα που δεν έχουν ξαναδεί. Αυτό συμβαίνει γιατί τα δεδομένα δεν έχουν ισχυρά μη γραμμικές σχέσεις και οι υποθέσεις για normality και ίσα co-variances ισχύουν.

Όσον αφορά την KNN, γενικά δεν κάνει υποθέσεις για τα δεδομένα, και συνεπώς για σετ στα οποία υπάρχουν περίπλοκες μη γραμμικές σχέσεις ενδεχομένως να είνια εξίσου ικανή στην πρόβλεψη. Εδώ βέβαια, οι επιδόσεις των μοντέλων είναι σαφώς χειρότερες.

4 Μοντέλα

4.1 Logistic Regression

Πίνακας 5: Logistic Refression coefficients και intercept για διαφορετικά thresholds.

Variable	**Threshold 0.4**	**Threshold 0.5**	**Threshold 0.6**
Intercept	2.6900×10^{-6}	1.8416×10^{-6}	2.6900×10^{-6}
x_1	-1.4814×10^{-3}	-1.4819×10^{-3}	-1.4814×10^{-3}
x_2	-1.3370×10^{-2}	-1.3414×10^{-2}	-1.3370×10^{-2}
x_3	8.0607×10^{-3}	6.2724×10^{-3}	8.0607×10^{-3}
x_4	3.1486×10^{-3}	3.7386×10^{-3}	3.1486×10^{-3}
x_5	6.5633×10^{-3}	8.7562×10^{-3}	6.5633×10^{-3}
x_6	3.1761×10^{-5}	3.4897×10^{-5}	3.1761×10^{-5}
x_7	3.9940×10^{-2}	3.4190×10^{-2}	3.9940×10^{-2}
x_8	-4.3707×10^{-3}	-5.1363×10^{-3}	-4.3707×10^{-3}
x_9	1.2228×10^{-2}	1.2879×10^{-2}	1.2228×10^{-2}
x_{10}	1.8919×10^{-3}	1.5298×10^{-3}	1.8919×10^{-3}
x_{11}	1.6494×10^{-2}	1.6056×10^{-2}	1.6494×10^{-2}
x_{12}	-1.3451×10^{-2}	-1.0350×10^{-2}	-1.3451×10^{-2}
x_{13}	7.1077×10^{-5}	4.2714×10^{-5}	7.1077×10^{-5}
x_{14}	-4.4294×10^{-10}	1.4012×10^{-11}	-4.4294×10^{-10}
x_{15}	-7.4266×10^{-3}	-6.1265×10^{-3}	-7.4266×10^{-3}
x_{16}	7.1989×10^{-4}	5.5155×10^{-4}	7.1989×10^{-4}
x_{17}	9.6267×10^{-3}	7.4146×10^{-3}	9.6267×10^{-3}

4.2 Linear Discriminant Analysis

Πίνακας 6: Linear Discriminant Analysis coefficients και intercepts για διαφορετικά thresholds.

Variable	**Threshold 0.4**	**Threshold 0.5**	**Threshold 0.6**
Intercept	220.8543	220.8543	220.8543
x_1	-1.1485×10^{-1}	-1.1485×10^{-1}	-1.1485×10^{-1}
x_2	-9.1135×10^{-3}	-9.1135×10^{-3}	-9.1135×10^{-3}
x_3	-3.4769×10^{-2}	-3.4769×10^{-2}	-3.4769×10^{-2}
x_4	-1.2630×10^{-4}	-1.2630×10^{-4}	-1.2630×10^{-4}
x_5	7.1530×10^{-3}	7.1530×10^{-3}	7.1530×10^{-3}
x_6	1.7716×10^{-5}	1.7716×10^{-5}	1.7716×10^{-5}
x_7	2.2966×10^{-2}	2.2966×10^{-2}	2.2966×10^{-2}
x_8	-1.9087×10^{-3}	-1.9087×10^{-3}	-1.9087×10^{-3}
x_9	2.3331×10^{-3}	2.3331×10^{-3}	2.3331×10^{-3}
x_{10}	1.2093×10^{-1}	1.2093×10^{-1}	1.2093×10^{-1}
x_{11}	-1.3324×10^{-4}	-1.3324×10^{-4}	-1.3324×10^{-4}
x_{12}	-5.5143×10^{-2}	-5.5143×10^{-2}	-5.5143×10^{-2}
x_{13}	1.3157×10^{-5}	1.3157×10^{-5}	1.3157×10^{-5}
x_{14}	-8.2513×10^{-10}	-8.2513×10^{-10}	-8.2513×10^{-10}
x_{15}	-3.1914×10^{-2}	-3.1914×10^{-2}	-3.1914×10^{-2}
x_{16}	6737.66198	6737.66198	6737.66198
x_{17}	477.30827	477.30827	477.30827