Technical University of Crete

Στατιστική Μοντελοποίηση και Αναγνώριση Προτύπων (ΤΗΛ311)

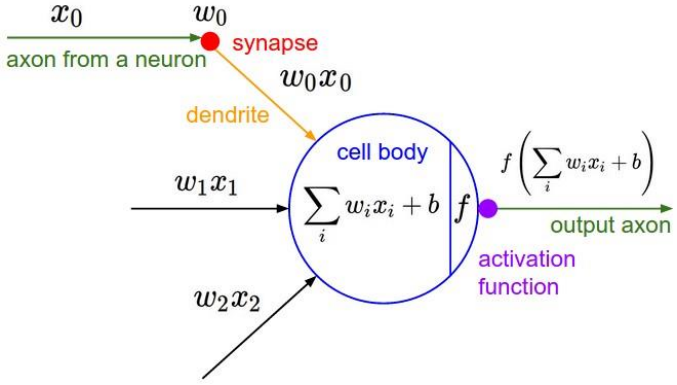
2Η ΣΕΙΡΑ ΑΣΚΗΣΕΩΝ

ΒΙΤΤΗΣ ΒΑΣΙΛΗΣ 2015030164

Θέμα 1: Αρχιτεκτονική Νευρωνικών δικτύων

1.1 Εισαγωγή

Η αρχιτεκτονική νευρωνικών δικτύων αποτελείται από έναν αριθμό επιπέδων (layers) τα οποία δομούνται από έναν αριθμό νευρώνων (neurons). Κάθε νευρώνας δέχεται ως είσοδο την σύναψη(πολλαπλασιασμό) του διανύσματος X_i με το αντίστοιχο βάρος W_i. Η διαδικασία την οποία εκτελεί κάθε νευρώνας είναι απλή και απεικονίζεται στην παρακάτω εικόνα στο κομμάτι του cell body.Το άθροισμα αυτό λαμβάνει υπόψη του τα βάρη και τα διανύσματα εισόδου καθώς επίσης και μια μετρική (bias) που καθορίζει το σημείο τομής της γραμμικής αυτής εξίσωσης με τον κατακόρυφο άξονα. Η έξοδος του νευρώνα είναι το αποτέλεσμα του αθροίσματος. Το πιο δύσκολο ίσως κομμάτι στην αρχιτεκτονική των νευρωνικών δικτύων είναι η εύρεση των κατάλληλων τιμών για τα βάρη. Έτσι σαν πρώτο βήμα κατά την διαδικασία αυτή πρέπει να ορίσουμε τον αριθμό των παραμέτρων (βάρη) που θα χρησιμοποιηθούν. Αποδεικνύεται εύκολα με απλή παρατήρηση ότι σε ένα fully connected neural network με δυο hidden layers ο αριθμός των βαρών είναι: Πλήθος βαρών $= I \cdot H1 + H1 \cdot H2 + H2 \cdot O$. Όπου I οι διαστάσεις στην είσοδο, H_i ο αριθμός των νευρώνων σε κάθε επίπεδο και O ο αριθμός των τερματικών κόμβων .



1.2 Διαδικασία και Αποτελέσματα

Άρα με βάση την παραπάνω ανάλυση προκύπτει ότι:

Πλήθος βαρών = I·H1 +H1·H2 +H2·O = 12288·100 + 100·100 + 100·10

=1.228.000 + 10.000 + 1.000

=1.239.000

Ο συνολικός αριθμός βαρών στο δίκτυο είναι 1.239.000.

Θέμα 2: Λογιστική Παλινδρόμηση: Αναλυτική εύρεση κλίσης (Gradient)

2.1 Εισαγωγή

Σε αυτή την άσκηση κάνουμε μια εκτίμηση μέσω της λογιστικής παλινδρόμησης. Κύριος στόχος μας είναι να βρούμε να κατάλληλες παραμέτρους για να κάνουμε καλύτερα την πρόβλεψη μας.

$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \cdots + \theta_n x_n$$

Για να το κάνουμε αυτό πρέπει να βλέπουμε παράλληλα το αποτέλεσμα των επιλογών μας για τις τιμές αυτές. Μέσω της γραμμικής παλινδρόμησης και της συνάρτησης κόστους cross-entropy μπορούμε να υπολογίσουμε το σφάλμα. Επίσης σκοπός μας είναι να υπολογίζουμε την κλίση της συνάρτησης.

2.2 Διαδικασία και Αποτελέσματα

2.2.1: Απόδειξη κλίσης σφάλματος j-στοιχείου

Γνωρίζουμε ότι η συνάρτηση λογιστικής παλινδρόμησης είναι:

$$h_\theta(x) = f(\theta^T x)$$

όπου $\boldsymbol{\theta}$=$[\theta 1, \theta 2, \cdots, \theta n]^{\mathrm{T}}$ είναι οι παράμετροι του γραμμικού μοντέλου και $f()$ είναι η λογιστική συνάρτηση που ορίζεται ως:

$$f(z) = \frac{1}{1 + e^{-z}}$$

Ορίζουμε ως την εκτίμηση της λογιστικής συνάρτησης της κλάσης y^i την μεταβλητή $\hat{y}^{(i)} = h_\theta(x^{(i)})$

Αντικαθιστώντας το $\hat{y}^{(i)}$ έχουμε:

$$J(\theta) = \frac{1}{m} \sum_{i=1}^{m} (-y^{(i)} \ln\left(h_\theta\left(x^{(i)}\right)\right) - \left(1 - y^{(i)}\right) \ln\left(1 - h_\theta\left(x^{(i)}\right)\right) \qquad (1)$$

Υπολογίζοντας το $h_\theta(x^{(i)})$ προκύπτει ότι έχουμε:

$$h_\theta\left(x^{(i)}\right) = \frac{1}{1 + e^{-\theta^T x(i)}}$$

Έτσι, συνεχίζοντας την προεργασία μας για να υπολογίσουμε το $J(\theta)$ υπολογίζουμε συνιστώσες που θα μας φανούν χρήσιμες όπως:

$$\ln\left(h_\theta\left(x^{(i)}\right)\right) = \ln\left(\frac{1}{1+e^{-\theta^T x(i)}}\right) = -\ln\left(1+e^{-\theta^T x(i)}\right)$$

$$\ln\left(1-\ h_\theta\left(\mathrm{x}^{(\mathrm{i})}\right)\right) = \ln\left(1-\ \frac{1}{1+\mathrm{e}^{-\theta^{\mathrm{T}}\mathrm{x}(\mathrm{i})}}\right)$$

$$= \ln\left(\mathrm{e}^{-\theta^{\tau}\,\mathrm{x}(\mathrm{i})}\right)\ -\ln\left(1+\mathrm{e}^{-\theta^{\mathrm{T}}\mathrm{x}(\mathrm{i})}\right) = -\theta^T x(i) - \ln\left(1+\mathrm{e}^{-\theta^{\mathrm{T}}\mathrm{x}(\mathrm{i})}\right)$$

Έτσι, έχοντας υπολογίσει όλα τα επιμέρους στοιχεία της $J(\theta)$, υπολογίζουμε την $J(\theta)$

Από την (1) έχουμε:¨

$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m}(-y^{(i)}\ln\left(\frac{1}{1+e^{-\theta^T x(i)}}\right) - \left(1-y^{(i)}\right)\ln\left(1-\ \frac{1}{1+e^{-\theta^T x(i)}}\right)\ =$$

$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m}(-y^{(i)}\cdot\left(-\ln\left(1+e^{-\theta^T x(i)}\right)\right) - \left(1-y^{(i)}\right)\ln\left(1-\ \frac{1}{1+e^{-\theta^T x(i)}}\right)\ =$$

$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m}(-y^{(i)}\cdot\left(-\ln\left(1+e^{-\theta^T x(i)}\right)\right) - \left(1-y^{(i)}\right)\cdot\left(-\theta^T x(i) - \ln\left(1+\mathrm{e}^{-\theta^{\mathrm{T}}\mathrm{x}(\mathrm{i})}\right)\right)\ =$$

$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m} y^{(i)}\cdot\ln\left(1+e^{-\theta^T x(i)}\right) + \left(1-y^{(i)}\right)\left(\ \theta^T x(i) + \ln\left(1+\mathrm{e}^{-\theta^{\mathrm{T}}\mathrm{x}(\mathrm{i})}\right)\right) =$$

$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m}\ln\left(\mathrm{e}^{-\theta^{\mathrm{T}}\mathrm{x}(\mathrm{i})}\left(1+e^{-\theta^T x(i)}\right)\right) -\ y^{(i)}\ \theta^T x(i)\ =$$

$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m}\ln\left(1+e^{\theta^T x(i)}\right) -\ y^{(i)}\cdot\theta^T\cdot x(i)\ =$$

Άρα παραγωγίζοντας την $J(\theta)$ ως προς θ , έχουμε:

$$\frac{dJ(\theta)}{d\theta} = \frac{1}{m}\sum_{i=1}^{m}\ln\left(1+e^{\theta^T x(i)}\right) - y^{(i)}\cdot\theta^T\cdot x(i)\quad =$$

$$= \frac{1}{m}\sum_{i=1}^{m}\left(\frac{x(i)+e^{\theta^T x(i)}}{1+e^{\theta^T x(i)}}\right) - y^{(i)}\cdot x(i)\ = \frac{1}{m}\sum_{i=1}^{m}\left(\frac{1}{1+e^{\theta^T x(i)}}\ -\ y^{(i)}\right)\cdot x(i)$$

Όμως γνωρίζουμε ότι το $\frac{1}{1+e^{\theta^T x(i)}} = h_\theta\left(x^{(i)}\right)$.Άρα

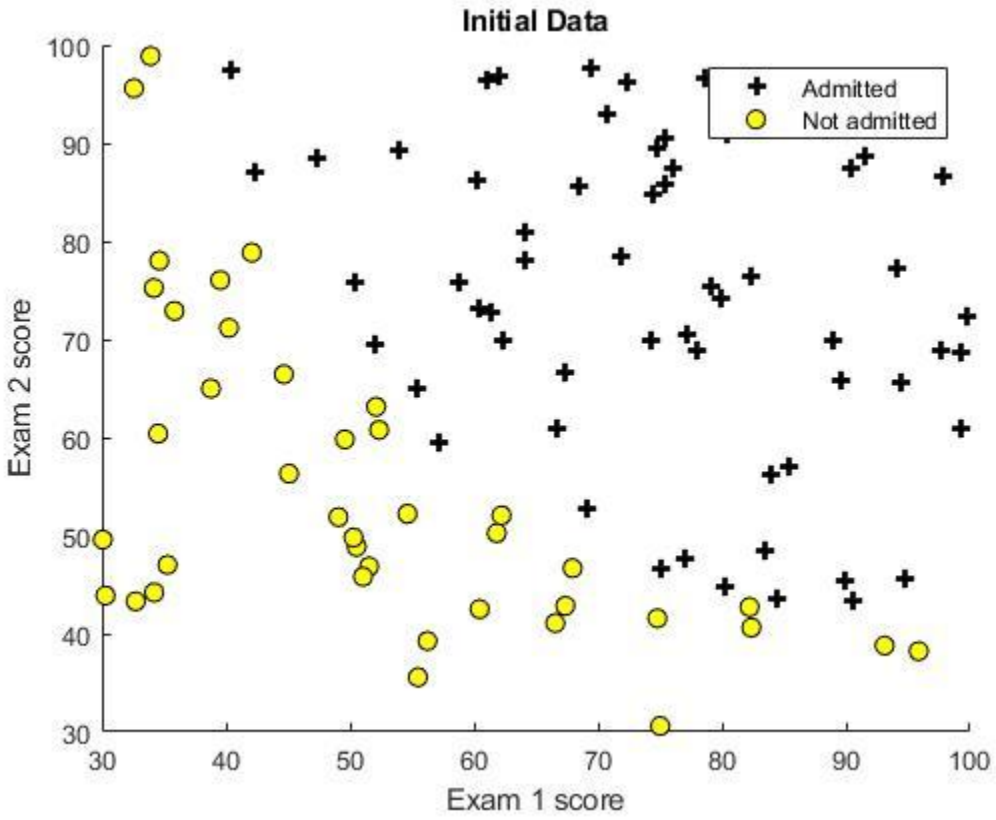
$$\frac{dJ(\theta)}{d\theta} = \frac{1}{m}\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right)\ -\ y^{(i)}\right)\cdot x(i)$$

2.2.2: Υλοποίηση Λογιστικής Παλινδρόμησης

Στην παρακάτω άσκηση εφαρμόζονται τα κάτωθι βήματα: Υλοποίηση της σιγμοειδούς συνάρτησης f(z), υλοποίηση της συνάρτησης cost function για τον υπολογισμό του κόστους αλλά και της κλίσης της

συνάρτησης ,βελτιστοποίηση των παραμέτρων , εύρεση του συνόρου απόφασης και τέλος την πρόβλεψη του ζητούμενου.

Αρχικά απεικονίσαμε τα δεδομένα μας όπως βλέπουμε και παρακάτω για να έχουμε μια πρώτη εικόνα, αυτό έγινε δυνατό με την συνάρτηση plotData.m



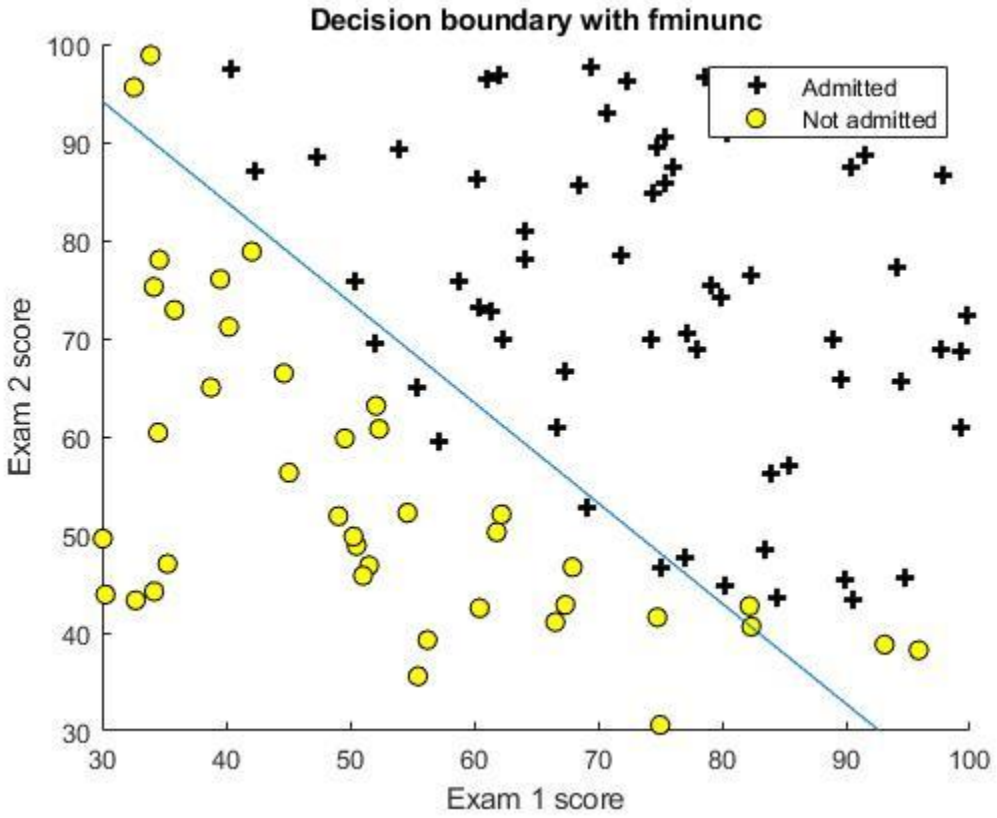
Εικόνα 1

Έπειτα υπολογίσαμε το σφάλμα με βάση τη συνάρτηση κόστους/σφάλματος (loss function), που ονομάζεται cross-entropy αφού πρώτα είχαμε κατασκευάσει την σιγμοειδή συνάρτηση. Πιο συγκεκριμένα υπολογίζουμε αρχικά την υπόθεση μας μέσω της σιγμοειδούς συνάρτησης και ύστερα υπολογίζουμε το κόστος και το διάνυσμα gradient βασιζόμενοι την υπόθεση μας.

Τα αποτελέσματα της συνάρτησης σφάλματος είναι πολύ λογικά καθώς μπορεί να πέσαμε στο πρόβλημα της συγκεκριμένης διαδικασίας, το οποίο είναι ότι μπορεί να βρεθήκαμε σε ένα τοπικό ελάχιστο το οποίο να μην ήταν το καλύτερο που μπορούσαμε να βρούμε.

Η αρχική τιμή του σφάλματος είναι φυσιολογικό να είναι υψηλή J= 0.693147, ενώ προκύπτει ότι η ευθεία που δημιουργείται έχει ως θ0 και θ1 και b ή αλλιώς το gradient διάνυσμα είναι περίπου [-0.10 ,-12.00921, -11.26284].

Έτσι, εφαρμόζοντας βελτιστοποίηση στις παραμέτρους και εφαρμόζοντας την συνάρτηση fminunc η οποία βρίσκει το τοπικό ελάχιστο μιας συνάρτησης πολλών παραμέτρων καταφέραμε πιο αποδεκτά αποτελέσματα.
Κόστος του θ με την χρήση της fminunc: 0.203498, ενώ το gradient διάνυσμα είναι [-25.161343 0.206232, 0.201472] .Το αποτέλεσμα είναι πολύ καλύτερο από πριν καθώς βλέπουμε όσον αφορά το κόστος ότι έχει μειωθεί αισθητά, έως και τρεις φορές πιο κάτω.



Εικόνα 1

Έχοντας άρα στην διάθεση μας τις μεταβλητές του theta ή αλλιώς τις συνιστώσες του decision boundary μπορούμε να κάνουμε μια πρόβλεψη για ένα καινούργιο δείγμα. Άρα για έναν μαθητή έχοντας γράψει στο πρώτο διαγώνισμα 45 και στο δεύτερο 85 η πιθανότητα να περάσει και να γίνει αποδεκτός είναι περίπου 0.776291 με ακρίβεια 0.89. Στην παραπάνω εικόνα είναι εύκολο να παρατηρήσουμε ότι όντως 11 από τα 100 δείγματα μας είναι λάθος ταξινομημένα.

Τέλος συμπεραίνουμε ότι η χρήση της σιγμοειδούς συνάρτησης σε συνδυασμό με την συγκεκριμένη συνάρτηση σφάλματος είναι μια καλή επιλογή για πρόβλεψη αλλά πρέπει να συνδυαστεί με την εύρεση του κατάλληλου τοπικού ελαχίστου που θα μας δώσει την βέλτιστη λύση.

Θέμα 3: Νευρώνες Sigmoid και SoftMax: Βελτιστοποίηση κατά προσέγγιση χρησιμοποιώντας SGD

3.1 Εισαγωγή

Σε αυτή την άσκηση μελετάμε τρόπους για να εκτιμήσουμε τιμές στις παραμέτρους του νευρώνα $W\ \varepsilon\ R^{2X1} και\ b\ \varepsilon\ R$ έτσι ώστε να κάνουμε πρόβλεψη για το αν έναν μαθητής με βάση του βαθμούς του σε δυο διαγωνίσματα θα επιλεγεί ή όχι. Θα συγκρίνουμε δύο συναρτήσεις ενεργοποίησης υλοποιώντας στο πρώτο μισό την λογιστική συνάρτηση (sigmoid) και στο δεύτερο μισό την συνάρτηση Softmax. Η πρώτη διαφορά είναι ότι η λογιστική συνάρτηση (sigmoid) έχει μια είσοδο και μια έξοδο ενώ η softmax έχει μια είσοδο και δυο εξόδους.

3.2 Διαδικασία και Αποτελέσματα

3.2.1: Sigmoid

Χρησιμοποιώντας την λογιστική συνάρτηση sigmoid προκύπτει ότι η πρόβλεψη είναι ότι θα γίνει admitted δηλαδή αποδοχή καθώς το binary prediction = 1 τις περισσότερες φορές ενώ παράλληλα η πιθανότητα πρόβλεψης είναι = 0.66190165. Άρα, συμπεραίνουμε ότι κατά 0.66 % με βάση τους βαθμούς 45 και 85 στο πρώτο και δεύτερο διαγώνισμα το σχολείο θα αποδεχθεί τον μαθητή.

Αρχικά κατά την υλοποίηση του σιγμοειδούς inference model πάνω στα αρχικά δεδομένα χρησιμοποιήσαμε ένα threshold στο 0.5 με βάση το οποίο παράγονται τα παραπάνω αποτελέσματα. Ύστερα από πειραματισμούς καταλήγουμε στο αυτονόητο δηλαδή ότι όσο πιο πολύ αυξάνουμε το threshold τόσο πιο δύσκολο είναι για την συνάρτηση μας να καταφέρει να ενεργοποιηθεί. Έτσι, παρατηρούμε ότι πάνω από την τιμή 0.67 το σύστημα μας τείνει να απορρίπτει τον φοιτητή ενώ κάτω από την τιμή 0.46 τείνει να τον αποδέχεται. Στις τιμές που βρίσκονται ενδιάμεσα παρατηρείται μια απροβλεψιμότητα καθώς κάποιες ενεργοποιείται άρα τον αποδέχεται και κάποιες όχι.

Τέλος, πειραματιζόμενοι με το μέγεθος των batches που επί της ουσίας είναι με μετρική που ορίζει με πόσα δείγματα θα δουλέψει ένα πέρασμα του αλγορίθμου πριν ανανεωθούν προέκυψε ότι την καλύτερη πιθανότητα πρόβλεψης την είχαν οι αριθμού από 13 έως 15 δίνοντας έως και 0.737650 % πιθανότητα σωστής πρόβλεψης. Ενώ με πολύ μικρές ή πολύ μεγάλες τιμές η πιθανότητα πρόβλεψης έπεφτε αισθητά. Ενώ, σχετικά με το learning rate παρατηρούμε ότι όταν είναι μικρό κάνουμε μικρά αλλά αργά βήματα προς τον στόχο μας αλλά μπορεί να κωλύσουμε σε ένα σημείο ελαχίστου άρα και πιο σίγουρα βήματα αλλά με μεγάλο κάνουμε μεγάλα και γρήγορα βήματα τα οποία μπορούν να μας απομακρύνουν από τον στόχο μας.

3.2.2: Softmax

Στην δεύτερη αυτή εκδοχή η λογιστική συνάρτηση ενεργοποίησης αντικαθίσταται από τη συνάρτηση SoftMax. Όπως γνωρίζουμε η συνάρτηση softmax μετατρέπει ένα διάνυσμα πραγματικών αριθμών σε ένα διάνυσμα από posteriori πιθανότητες και οι τιμές στην έξοδοι της δεν είναι δυαδικές αλλά ανήκουν στο φάσμα ανάμεσα το 0 και το 1. Πρώτη αρχική διαφορά είναι ότι δεν έχουμε κάποιο threshold αλλά έχουμε τις εκείνη την τιμή με την μεγαλύτερη πιθανότητα. Επίσης, παρατηρούμε ότι οι πιθανότητες των διάφορων βαθμών αθροίζουν στο 1.Άρα, βλέποντας τις επιμέρους δυο πιθανότητες μπορούμε να πούμε ότι ο βαθμός παίζει σημαντικότερο ρόλο στο να γίνει έναν φοιτητής admitted καθώς η πιθανότητα αυτού είναι μεγαλύτερη. Ενδεικτικά αναφέρουμε ότι το epoch_loss = 0.296960550 ενώ το Binary Prediction =

[1] δηλαδή αποφασίζει ότι είναι admitted και τα Prediction Probabilities = 0.47353104 0.5264689 για τις 2 εισόδους με Accuracy of Training Samples = 0.89.

Θέμα 4: Feature Selection – Classification – Cross Validation – Overfitting

4.1 Εισαγωγή

Σε αυτή την άσκηση υλοποιούμε την επίδοση του συστήματος με την τεχνική το cross validation leave-one-out. Ο λόγος που επιλέγουμε αυτή την τεχνική σε σχέση με τις υπόλοιπες (k-fold, leave-p-out) είναι ο περιορισμένος αριθμός δειγμάτων. Σκοπός μας είναι να ελέγξουμε την συσχέτιση των χαρακτηριστικών με τα labels, δηλαδή ποια χαρακτηριστικά έχουν μεγαλύτερη βαρύτητα και μεγαλύτερη συσχέτιση με το αν ένας άνθρωπος έχει αυτισμό ή όχι. Για να επιτύχουμε αυτό τον σκοπό στην συγκεκριμένη άσκηση κάνουμε χρήση του συντελεστή συσχέτισης Pearson (Pearson's correlation coefficient).

Άρα, συμπεραίνουμε ότι είμαστε ένα βήμα πριν την υλοποίηση του classifier καθώς χρησιμοποιούμε τον συντελεστή Pearson στην διαδικασία του feature selection.Ενώ, στο cross-validation επί της ουσίας καλούμε επαναληπτικά τον classifier μας. Η διαφοροποίηση που προκύπτει κατά την διαδικασία του cross-validation είναι ποιο κομμάτι των αρχικών μας δεδομένων θα τεθεί προς training και ποιο κομμάτι προς testing. Έτσι, έχει νόημα να μελετήσουμε αν το feature selection θα γίνει με ή χωρίς το ένα sample που αφήνουμε εκτός κατά το leave-one-out cross validation method.

Παρατήρηση: Το γεγονός ότι τα δεδομένα μας είναι κανονικοποιημένα τυχαία δεν μας βοηθάει να προσεγγίσουμε το πρόβλημα μας με ακρίβεια καθώς όπως είναι προφανές δεν γίνεται να είναι καλά διαχωρισμένα στις δυο κλάσεις. Έτσι και τα παρακάτω ποσοστά στην ακρίβεια δεν θα έχουν τόσο μεγάλη διαφοροποίηση. Δεν είναι αλγοριθμικό πρόβλημα αλλά πρόβλημα που παράγεται από την μορφή των δεδομένων μας.

4.2 Διαδικασία και Αποτελέσματα

4.2.1 Classify without feature selection

Στο πρώτο κομμάτι της άσκησης δεν κάνουμε feature selection και χρησιμοποιούμε και τα 1000 χαρακτηριστικά για να κάνουμε train τον classifier μας. Αυτό έχει ως αποτέλεσμα προφανώς να ρίχνει την απόδοση της πρόβλεψης καθώς κάνει overfitting και αυξάνοντας την πολυπλοκότητα των πράξεων άρα και τις διαστάσεις χάνουμε σε απόδοση. Classify without feature selection accuracy :0.6

4.2.2 Classify with feature selection inside the cross -validation

Απόρια της παραπάνω παρατήρησης είναι ότι το γεγονός ότι μειώσαμε τις διαστάσεις άρα και τα features από 1000 στα 100 δεν έχει κάποιο αντίκτυπο το οποίο δεν είναι λογικό. Αυτή η μέθοδος ανάμεσα στις τρείς είναι η πιο σωστή καθώς το feature selection δεν λαμβάνει υπόψη του το testing data άρα το testing έχει νόημα να υπάρχει καθώς είναι πραγματικά ένα νέο δεδομένα για το σύστημα Η ακρίβεια σε αυτή την μέθοδο είναι :0.52

4.2.3 Classify with feature selection outside the cross-validation: accuracy :1

Τέλος, έχουμε την μέθοδο με το feature selection να γίνεται έξω από το loop του cross-validation και να λαμβάνει υπόψη του όλα τα 25 samples. Άρα επί της ουσίας το cross-validation ως testing method δεν έχει βάση καθώς δεν ελέγχουμε κάποιο καινούργιο sample μιας και το testing sample το έχουμε λάβει υπόψη μας κατά την διαδικασία του training.Άρα κάνουμε overfitting το όποιο δεν θέλουμε.