Πανεπιστήμιο Δυτικής Μακεδονίας

Σχολή Οικονομικών Επιστημών

Τμήμα Διοικητικής Επιστήμης και Τεχνολογίας

Μάθημα Αναλυτική Επιχειρήσεων

Η’ Εξάμηνο

Εργασία εξαμήνου

Εδώ μπαίνει ονοματεπώνυμο

Απρίλιος 2023

Περιεχόμενα

[1. Ερώτημα 1 5](#_Toc130563125)

[2. Ερώτημα 2 6](#_Toc130563126)

[3. Ερώτημα 3 7](#_Toc130563127)

[4. Ερώτημα 4 8](#_Toc130563128)

Πίνακες

[Πίνακας 1Μέτρα θέσης και διασποράς της μεταβλητής Diagnosis 5](#_Toc130561953)

[Πίνακας 2Πίνακας πιθανοτήτων της μεταβλητής X\_test 8](#_Toc130561954)

Εικόνες

[Εικόνα 1Μέση Τιμή και Διακύμανση της μεταβλητής Diagnosis 7](#_Toc130562061)

# Ερώτημα 1

Στο ερώτημα πρώτο στόχος είναι να διαβαστεί το σύνολο δεδομένων από το αρχείο wdbc.data. Μετέπειτα, να μετατραπεί ο κατηγορικός τύπος σε binary, 1 και 0 όπως και έγινε.

Οι παρακάτω γραμμές κώδικα σε Python 3.11. αναπαριστούν την εφαρμογή του πρώτου ερωτήματος.

Ουσιαστικά, διαβάσαμε τα δεδομένα και τα αποθηκεύσαμε σε μία μορφή δομή δεδομένων data frame όπου φορτώθηκε από την βιβλιοθήκη pandas.

data = pd.read\_csv('wdbc.data', sep=",", header=None)

Μετέπειτα έγινε αντιστοίχιση των στηλών με τα ονόματα των στηλών όπως βρέθηκαν στην ιστοσελίδα <https://archive-beta.ics.uci.edu/dataset/17/breast+cancer+wisconsin+diagnostic>.

Δηλαδή οι στήλες αντιστοιχήθηκαν με τα χαρακτηριστικά .

data.columns=['ID', 'Diagnosis', 'radius1','texture1', 'perimeter1', 'area1', 'smoothness1', 'compactness1', 'concavity1', 'concave\_points1', 'symmetry1','fractal\_dimension1', 'radius1','texture2','perimeter2', 'area2','smoothness2', 'compactness2', 'concavity2', 'concave\_points2', 'symmetry2', 'fractal\_dimension2', 'radius3', 'texture3', 'perimeter3', 'area3', 'smoothness3', 'compactness3', 'concavity3','concave\_points3', 'symmetry3', 'fractal\_dimension3']

Στη συνέχεια έγινε αντικατάσταση μόνο του χαρακτηριστικού Diagnosis από Μ και Β σε 1 και 0 όπως λέει το πρώτο ερώτημα. Έτσι, μετατράπηκε η κατηγορηματική μεταβλητή σε binary.

data['Diagnosis'].replace(['M', 'B'], [1, 0], inplace=True)

# Ερώτημα 2

Στο δεύτερο ερώτημα στόχος είναι να βρεθεί η μέση τιμή, η χαμηλότερη τιμή, η υψηλότερη τιμή, η τυπική απόκλιση και η διακύμανση του δείγματος. Σύμφωνα με το εξής κομμάτι κώδικα:

data['Diagnosis'].describe()

mean = data['Diagnosis'].describe().loc['mean']

std = data['Diagnosis'].describe().loc['std']

print(data['Diagnosis'].describe())

Δηλαδή ουσιαστικά την πληροφορία του ερωτήματος 2 την βρίσκουμε από την μέθοδο describe(). Οπότε, στον Πίνακας 1 παρουσιάζονται η μέση τιμή, η τυπική απόκλιση, η διακύμανση, η μικρότερη και η μεγαλύτερη τιμή της μεταβλητής Diagnosis. Το πλήθος των τιμών είναι 569, η μέση τιμή ισούται με 0.372583, η τυπική απόκλιση είναι 0.483918, η διακύμανση είναι 0.0234176631, η μικρότερη τιμή είναι 0, η μεγαλύτερη τιμή είναι 1. Το 25% των τιμών ανήκει στην τιμή 0, το 50% των τιμών ανήκουν στο 0, το 75% των τιμών ανήκουν στην τιμή 1.

Πίνακας 1Μέτρα θέσης και διασποράς της μεταβλητής Diagnosis

|  |  |
| --- | --- |
| Count | 569.000000 |
| Mean | 0.372583 |
| Std | 0.483918 |
| Variance | 0.0234176631 |
| Min | 0.000000 |
| Max | 1.000000 |
| 25% | 0.000000 |
| 50% | 0.000000 |
| 75% | 1.000000 |

# Ερώτημα 3

Σε αυτό το ερώτημα στόχος είναι το ραβδόγραμμα μεταξύ της μέσης τιμής και της διακύμανσης της μεταβλητής Diagnosis. Στην Εικόνα 1 παρουσιάζεται το ραβδόγραμμα της μέσης τιμής και διακύμανσης. Η μέση τιμή είναι στο 0.35 και η διακύμανση είναι στο 0.23. Το χρώμα επιλογής στην συνάρτηση είναι το maroon με πλάτος 0.3. Παρακάτω, εμφανίζεται ο κώδικας για να γίνει το ραβδόγραμμα του ερωτήματος 3.

dataMeanAndVariance = {'Mean': mean, 'Var': variance}

meanvar = list(dataMeanAndVariance.keys())

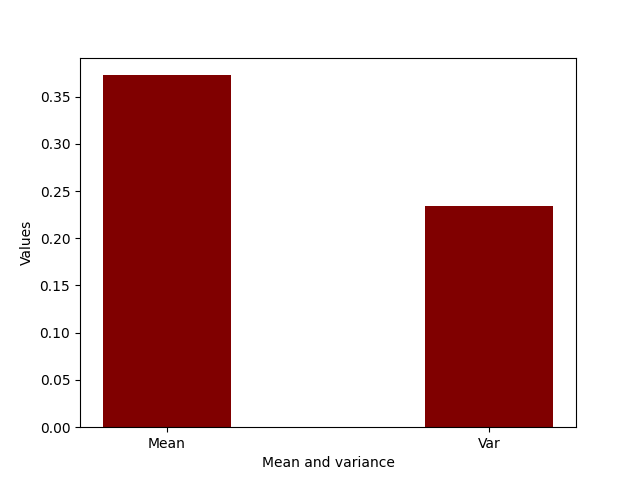
values = list(dataMeanAndVariance.values())

plt.bar(meanvar, values, color='maroon', width=0.4)

plt.xlabel("Mean and variance")

plt.ylabel("Values")

plt.show()



Εικόνα Μέση Τιμή και Διακύμανση της μεταβλητής Diagnosis

# Ερώτημα 4

Στο τέταρτο ερώτημα στόχος είναι να βάλουμε ένα μοντέλο μηχανικής μάθησης LogisticRegression() όπου σε αυτό για μεταβλητή y βάζουμε το Diagnosis και για μεταβλητές X βάζουμε όλες τις υπόλοιπες μεταβλητές εκτός της Diagnosis και το ID όπως μας λέει η εκφώνηση της άσκησης. Ο παρακάτω κώδικας μάς δημιουργεί το μοντέλο μηχανικής μάθησης. Για μεταβλητή y θεωρούμε την μεταβλητή Diagnosis και για μεταβλητή X βάζουμε όλες τις υπόλοιπες μεταβλητές. Μετέπειτα, χωρίζουμε όλο το υπόλοιπο σύνολο δεδομένων σε train και test data sets. Θεωρούμε το 80% να είναι το train set και σε 20% ελέγχουμε τις τιμές και το σκορ στο test set. Μετέπειτα, ορίζουμε το μοντέλο μηχανικής μάθησης να είναι το LogisticRegression με μέγιστο αριθμό επαναλήψεων να είναι οι 5000. Στη συνέχεια προσαρμόζουμε το μοντέλο μηχανικής μάθησης μας με τις τιμές X\_train, y\_train. Και μετέπειτα, κάνουμε προβλέψεις με τις μεταβλητές X\_test, y\_test. Το σκορ ακρίβειας του μοντέλου μηχανικής μάθησης μας είναι 0.9736842105263158. Μετέπειτα, εκτυπώνουμε τις πιθανότητες μας δηλαδή στο X\_test όπως φαίνεται στον Πίνακας 2. Με αυτόν τον τρόπο εφαρμόζεται το μοντέλο μηχανικής μάθησης μας στο σύνολο δεδομένων μας.

y = data['Diagnosis']

X = data.loc[:, ~data.columns.isin(['ID', 'Diagnosis'])]

#Άρα έχουμε δύο μεταβλητές την Χ και την y.

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2)

model = LogisticRegression(max\_iter=5000)

model.fit(X\_train, y\_train)

model.predict(X\_test)

model.score(X\_test, y\_test)

print(model.score(X\_test, y\_test))

model.predict\_proba(X\_test)

print(model.predict\_proba(X\_test))

Πίνακας Πίνακας πιθανοτήτων της μεταβλητής X\_test

[[9.98629400e-01 1.37059964e-03]

[9.89436772e-01 1.05632280e-02]

[8.76066132e-01 1.23933868e-01]

[9.89139898e-01 1.08601021e-02]

[9.97445450e-01 2.55454983e-03]

[1.08644310e-02 9.89135569e-01]

[5.45486836e-04 9.99454513e-01]

[9.98955185e-01 1.04481514e-03]

[8.84405336e-07 9.99999116e-01]

[9.97585348e-01 2.41465202e-03]

[9.96238542e-01 3.76145753e-03]

[9.99998685e-01 1.31528184e-06]

[5.68662464e-06 9.99994313e-01]

[9.66959537e-01 3.30404626e-02]

[9.99627362e-01 3.72637715e-04]

[9.99627668e-01 3.72331971e-04]

[9.99997988e-01 2.01175928e-06]

[9.89800044e-01 1.01999556e-02]

[1.40335403e-02 9.85966460e-01]

[9.99990332e-01 9.66772285e-06]

[9.47496364e-01 5.25036359e-02]

[9.99506641e-01 4.93359267e-04]

[1.11468612e-11 1.00000000e+00]

[9.49204199e-01 5.07958009e-02]

[9.91160457e-01 8.83954273e-03]

[9.99133950e-01 8.66049605e-04]

[8.59387212e-07 9.99999141e-01]

[9.84485771e-01 1.55142290e-02]

[9.99995691e-01 4.30922878e-06]

[9.99003760e-01 9.96239630e-04]

[3.67042351e-02 9.63295765e-01]

[4.14840662e-02 9.58515934e-01]

[9.99981342e-01 1.86583777e-05]

[3.86053974e-04 9.99613946e-01]

[9.89527892e-01 1.04721083e-02]

[8.91538734e-01 1.08461266e-01]

[9.99500097e-01 4.99902897e-04]

[9.99534466e-01 4.65533847e-04]

[2.03656692e-02 9.79634331e-01]

[4.44089210e-16 1.00000000e+00]

[2.22779128e-10 1.00000000e+00]

[2.32372690e-03 9.97676273e-01]

[9.99088236e-01 9.11763535e-04]

[9.19688975e-01 8.03110251e-02]

[9.96804513e-01 3.19548654e-03]

[7.44617539e-01 2.55382461e-01]

[9.99181495e-01 8.18504748e-04]

[9.80291423e-01 1.97085772e-02]

[9.97864924e-01 2.13507557e-03]

[9.99961065e-01 3.89351131e-05]

[3.36368274e-02 9.66363173e-01]

[9.37833144e-01 6.21668557e-02]

[7.78583306e-01 2.21416694e-01]

[9.73289013e-01 2.67109874e-02]

[5.82901122e-02 9.41709888e-01]

[9.99630723e-01 3.69276828e-04]

[3.90798505e-14 1.00000000e+00]

[9.94843100e-01 5.15690013e-03]

[7.38066727e-01 2.61933273e-01]

[9.99345223e-01 6.54776655e-04]

[9.98587261e-01 1.41273907e-03]

[9.89614002e-01 1.03859979e-02]

[9.99895350e-01 1.04650243e-04]

[9.63350864e-01 3.66491363e-02]

[7.87901666e-01 2.12098334e-01]

[9.97754846e-01 2.24515375e-03]

[3.16096038e-11 1.00000000e+00]

[9.95658718e-01 4.34128182e-03]

[9.99987541e-01 1.24590130e-05]

[9.94913116e-01 5.08688383e-03]

[1.81815599e-05 9.99981818e-01]

[4.53792559e-12 1.00000000e+00]

[9.96405623e-01 3.59437716e-03]

[9.76787194e-01 2.32128061e-02]

[9.99253364e-01 7.46636019e-04]

[9.99286889e-01 7.13110649e-04]

[9.97767796e-01 2.23220395e-03]

[4.84057239e-13 1.00000000e+00]

[9.58902014e-01 4.10979863e-02]

[9.82299194e-01 1.77008064e-02]

[2.44249065e-15 1.00000000e+00]

[9.99946529e-01 5.34707834e-05]

[1.84150473e-11 1.00000000e+00]

[1.17313311e-05 9.99988269e-01]

[2.35772104e-03 9.97642279e-01]

[9.72389524e-01 2.76104763e-02]

[9.78270767e-01 2.17292329e-02]

[1.44499231e-02 9.85550077e-01]

[9.99515148e-01 4.84851967e-04]

[8.86023472e-01 1.13976528e-01]

[8.69280764e-01 1.30719236e-01]

[9.99289279e-01 7.10721119e-04]

[9.86164893e-01 1.38351070e-02]

[9.59676063e-01 4.03239372e-02]

[8.33756166e-01 1.66243834e-01]

[9.96600605e-01 3.39939456e-03]

[9.83721830e-01 1.62781697e-02]

[9.97621616e-01 2.37838354e-03]

[1.94044762e-01 8.05955238e-01]

[9.98420423e-01 1.57957737e-03]

[6.19515232e-06 9.99993805e-01]

[9.96425942e-01 3.57405849e-03]

[9.99889708e-01 1.10292205e-04]

[9.99888319e-01 1.11681347e-04]

[3.33066907e-15 1.00000000e+00]

[9.94328453e-01 5.67154664e-03]

[4.51806299e-01 5.48193701e-01]

[9.75769020e-01 2.42309797e-02]

[9.85355249e-01 1.46447506e-02]

[9.92921165e-01 7.07883488e-03]

[9.99333503e-01 6.66496642e-04]

[9.98494255e-01 1.50574522e-03]

[9.99710099e-01 2.89900775e-04]

[8.12726330e-03 9.91872737e-01]]