MELANOMA MACHINE LEARNING

Σαλτερής Γεώργιος 2136
Μουλόπουλος Αντώνιος 2104

Αποτελεσματα απο την εκπαίδευση διαφορων clasifiers που προβλεπουν αν καποιος εχει μελανομα η όχι. Οι classifier που χρησιμοποιουμε είναι οι Random Forest, Linear Regression,SVM Linear,Linear Logistic Regression,OLS, GaussianNB, Decision Tree και Perceptron.

Για το παρακατω πειραμα κρατήσαμε ολα τα snp με **pvalue <= 0.0001** από το αρχείο **assoc** που πήραμε απο το **plink.** Το assoc αρχείο είναι της μορφής:

CHR	**Chromosome code**
SNP	**Variant identifier**
BP	**Base-pair coordinate**
A1	**Allele 1 (usually minor)**
F_A	**Allele 1 frequency among cases**
F_U	**Allele 1 frequency among controls**
A2	**Allele 2**
CHISQ	**Allelic test chi-square statistic. *Not present with 'fisher'/'fisher-midp' modifier.***
P	**Allelic test p-value**
OR	**odds(allele 1 | case) / odds(allele 1 | control)**

Για να στήσουμε τον πινακα **ασθενείς-features,** όπου feature είναι ένα snp, κωδικοποιούμε το κάθε snp σε 0 ή 1 ή 2. Για να ξεκινήσουμε την κωδικοποίηση των snps όπως είπαμε και πριν πρέπει πρώτα να εξάγουμε το lgen αρχείο των ασθενών με τη χρήση του plink. Το lgen αρχείο είναι ξεχωριστό για κάθε χρωμόσωμα και περίεχει το snp του ασθενή με τα allele του. Είναι της μορφής:

1. **Family ID**
2. **Within-family ID**
3. **Variant identifier**
4. **Allele 1**
5. **Allele 2**

Το snp κάθε ασθένη παίρνει τον κωδικό 0 αν το allele1 **και** το allele2 απο το lgen αρχείο είναι διαφορετικο από το allele1(A1) του assoc αρχείου, παίρνει τον κωδικό 1 αν **μόνο** ένα από τα allele1 και allele2 απο το lgen αρχείο είναι ίδια με το allele1(A1) του assoc αρχείου και τέλος παίρνει τον κωδικό 2 αν το allele1 **και** το allele2 απο το lgen αρχείο είναι ίδια με το allele1(A1) του assoc αρχείου.

Τα snps με **pvalue <= 0.0001** είναι 5415 για να αποφύγουμε το **overfitting** μειώνουμε τα snps με τον εξής τρόπο, τον οποίο σκεφτήκαμε μόνοι μας. Αφού στήσουμε τον πίνακα με τους ασθενείς και τα χαρακτηριστικά τους, στη συνέχεια βρίσκουμε τον πίνακα με το **cosine similarity** μεταξύ των snps χρησιμοποιώντας την εντολή **metrics.pairwise.cosine_similarity.** Ορίζουμε έναν counter για κάθε snp. Αρχικά σκεφτήκαμε να κρατήσουμε τα snps που δεν είναι και πολυ όμοια μεταξύ τους συύμφωνα με την κωδικοποίηση που αναφέραμε και πιο πάνω. Έτσι κρατήσαμε τα snps που ειχαν ομοιότητα μεγαλύτερη ή ίση με 0,3 και μικότερη ή ίση με 0,6. Τα snps από αυτήν τη περιμέναμε. Αυτό ίσως να συνέβη, το ότι οι classifiers δεν έκαναν καλή δουλειά, επειδή κρατούσαμε στήλες που ήταν αρκετά ανόμοιες μεταξύ τους με αποτέλεσμα οι classifiers να μπορούν να πάρουν αποφάσεις μόνο για συγκεκριμένες περιπτώσεις.

Το επόμενο βήμα ήταν να κρατήσουμε όλα τα snps που δεν ανήκουν στην παραπάνω περιοχή ομοιότητας, δηλαδή να έχουν ομοιότητα μικρότερη απο 0,3 και μεγαλύτερη από 0,6 . Από αυτά τα snps αφαιρέσαμε τα snps που έχουν ομοιότητα 1 με κάποιο άλλο, δηλαδή έχουν ακριβώς την ίδια κωδικοποίηση. Επίσης κρατήσαμε τα snps που είναι μικρότερα απο το 10% των snps που ανήκουν στην περιοχή $0.3 \leq x \leq 0.6$. Αυτό το κάναμε για να έχουμε snps που να καλύπτουν όσες το δυνατόν περισσότερες περιπτώσεις. Τα snps από αυτή την διαδικασία ήταν 2563 και τα αποτελέσματα φαίνονται στον **πίνακα2.**

Κάναμε και μια τρίτη δοκιμή, επιλέξαμε 2563 snps με τυχαίο τρόπο που δεν ανήκουν στα snps που επιλέξαμε στη πιο πάνω διαδικασία και τα αποτελέσματα φαίνονται στον **πίνακα3.**

Επίσης κάναμε και μια δοκιμή και με όλα τα features που είναι 5415. Τα αποτελέσματα αυτής της δοκιμής φαίνονται στον **πίνακα4.** Προσοχή, το να επιλέξουμε όλα τα snps μπορεί να οδηγήσει σε **overfitting** όπως είπαμε και πιο πάνω**.**

Για τη μείωση των features χρησιμοποιήσαμε και 2 έτοιμες υλοποιήσεις της python. Η πρώτη υλοποίηση χρησιμοποιεί την τεχνική **removing features with low variance,** η οποία αφαιρεί όλα τα features με βάση το threshold που έχουμε ορίσει, αν δεν ορίσουμε κάποιο threshold αφαιρεί τα feature που έχουν την ίδια τιμή σε όλο το δείγμα. Τα αποτελέσματα αυτής της τεχνικής φαίνονται στον **πίνακα5** με 1420 χαρακτηριστικά. Η δεύτερη υλοποίηση χρησιμοποιεί την τεχνική **univariate feature selection.** Αυτή η τεχνική κρατάει τα k-καλύτερα features, το k το ορίζουμε εμείς. Τα αποτελέσματα αυτής της τεχνικής φαίνονται στον **πίνακα6.**

Η επιλογή των snps εξαρτάται κάθε φορά από το πλήθος των ασθενών που θα χρησιμοποιήσουμε για εκπαίδευση.

Οι ασθενεις μας ειναι στο συνολο 4980. Απο αυτούς οι 3962 δεν εχουν την ασθενεια**(control = 0)** και οι υπόλοιποι 1018 εχουν την ασθενεια**(case = 1)**.

Για κάθε classifier χωρίζουμε τα δεδομένα που έχουμε σε 10 σύνολα που είναι ξένα μεταξύ τους. Άρα τρέχουμε κάθε classifier δέκα φορές και κάθε φορά χρησιμοποιούμε τα 9 σύνολα ως δεδομένα εκπαίδευσης και το 1 σύνολο ως test για να δούμε την ικανότητα γενίκευση του classifier. Τα αποτελέσματα που φαίνονται στους παρακάτω πίνακες είναι ο μέσος όρος των αποτελεσμάτων που πήραμε από τις 10 φορές που τρέξαμε τον clasifier. Το Cross validation είναι το ποσοστό επιτυχιας της πρόβλεψης. Το recall υπολογίζεται από τον τύπο **true positives / (true positives + false negatives)** και το precision από τον τύπο **true positives / (true positives + false positives). True positives** είναι η πρόβλεψη του classifier για κάποιον ασθενή ότι έχει την ασθενεια και πράγματι την έχει. **False positives** είναι η πρόβλεψη του classifier για κάποιον ασθενή ότι έχει την ασθενεια και στην πραγματικότητα δεν την έχει. **False negatives** είναι η πρόβλεψη του classifier για κάποιον ασθενή ότι δεν έχει την ασθενεια και στην πραγματικότητα την έχει.

	Cross validation	AUC	recall	precision	F_Measure
Random Forest	**0,82**	**0,57**	**0,15**	**0,93**	**0,27**
Linear Regression	**0,86**	**0,74**	**0,53**	**0,77**	**0,63**
SVM(kernel = linear)	**0,87**	**0,78**	**0,63**	**0,73**	**0,68**
Linear Logistic Regression	**0,88**	**0,78**	**0,62**	**0,74**	**0,68**
Perceptron	**0,92**	**0,90**	**0,88**	**0,78**	**0,82**
OLS	**0,88**	**0,75**	**0,54**	**0,80**	**0,65**
Naive Bayes Gaussian	**0,93**	**0,87**	**0,76**	**0,89**	**0,82**
Decision Tree	**0,78**	**0,62**	**0,56**	**0,44**	**0,39**

Πίνακας 1

	Cross validation	AUC	recall	precision	F_Measure
Random Forest	**0,84**	**0,59**	**0,18**	**0,99**	**0,30**
Linear Regression	**0,87**	**0,72**	**0,48**	**0,82**	**0,6**
SVM(kernel = linear)	**0,96**	**0,93**	**0,86**	**0,99**	**0,92**
Linear Logistic Regression	**0,96**	**0,91**	**0,83**	**0,98**	**0,90**
Perceptron	**0,99**	**0,99**	**0,99**	**0,99**	**0,99**
OLS	**0,91**	**0,79**	**0,60**	**0,93**	**0,72**
Naive Bayes Gaussian	**0,96**	**0,97**	**0,99**	**0,85**	**0,91**
Decision Tree	**0,79**	**0,64**	**0,38**	**0,52**	**0,44**

Πίνακας 2

	Cross validation	AUC	recall	precision	F_measure
Random Forest	**0,80**	**0,52**	**0,05**	**0,88**	**0,1**
Linear Regression	**0,89**	**0,77**	**0,58**	**0,80**	**0,67**
SVM(kernel = linear)	**0,89**	**0,78**	**0,63**	**0,74**	**0,68**
Linear Logistic Regression	**0,88**	**0,76**	**0,57**	**0,79**	**0,66**
Perceptron	**0,92**	**0,85**	**0,72**	**0,91**	**0,80**
OLS	**0,89**	**0,78**	**0,59**	**0,85**	**0,70**
Naive Bayes Gaussian	**0,92**	**0,83**	**0,67**	**0,93**	**0,78**
Decision Tree	**0,75**	**0,58**	**0,32**	**0,36**	**0,33**

Πίνακας 3

	Cross validation	AUC	recall	precision	F_measure
Random Forest	**0,81**	**0,53**	**0,07**	**0,98**	**0,13**
Linear Regression	**0,86**	**0,72**	**0,51**	**0,71**	**0,59**
SVM(kernel = linear)	**0,95**	**0,91**	**0,83**	**0,86**	**0,89**
Linear Logistic Regression	**0,94**	**0,89**	**0,80**	**0,94**	**0,86**
Perceptron	**0,99**	**0,99**	**0,99**	**0,99**	**0,99**
OLS	**0,92**	**0,81**	**0,64**	**0,96**	**0,77**
Naive Bayes Gaussian	**0,95**	**0,97**	**0,99**	**0,83**	**0,90**
Decision Tree	**0,78**	**0,62**	**0,34**	**0,46**	**0,39**

Πίνακας 4

	Cross validation	AUC	recall	precision	F_measure
Random Forest	**0,79**	**0,50**	**0,08**	**0,70**	**0,01**
Linear Regression	**0,81**	**0,69**	**0,48**	**0,55**	**0,51**
SVM(kernel = linear)	**0,82**	**0,73**	**0,56**	**0,58**	**0,57**
Linear Logistic Regression	**0,84**	**0,74**	**0,57**	**0,62**	**0,59**
Perceptron	**0,86**	**0,72**	**0,49**	**0,75**	**0,59**
OLS	**0,81**	**0,69**	**0,48**	**0,56**	**0,52**
Naive Bayes Gaussian	**0,66**	**0,63**	**0,58**	**0,32**	**0,41**
Decision Tree	**0,70**	**0,54**	**0,27**	**0,27**	**0,27**

Πίνακας 5

	Cross validation	AUC	recall	precision	F_measure
Random Forest	**0,84**	**0,61**	**0,22**	**0,96**	**0,36**
Linear Regression	**0,91**	**0,82**	**0,66**	**0,87**	**0,75**
SVM(kernel = linear)	**0,95**	**0,91**	**0,82**	**0,96**	**0,89**
Linear Logistic Regression	**0,95**	**0,89**	**0,80**	**0,97**	**0,88**
Perceptron	**0,99**	**0,99**	**0,99**	**0,99**	**0,99**
OLS	**0,91**	**0,82**	**0,66**	**0,89**	**0,75**
Naive Bayes Gaussian	**0,93**	**0,92**	**0,90**	**0,81**	**0,85**
Decision Tree	**0,80**	**0,65**	**0,41**	**0,51**	**0,46**

Πίνακας 6