#### ΙΟΝΙΟ

#### ΠΑΝΕΠΙΣΤΗΜΙΟ

#### ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ – Αποθηκεσ και εξοριξη δεδομενων

# απαλλακτικη ΕΡΓΑΣΙΑ

**Ονοματεπώνυμο: Τασιόπουλος Μανώλης**

**ΑΜ : Π2015046**

**Διδάσκοντες : Θέμης Έξαρχος – Κάτια Κερμανίδου**

**Αριθμός σελίδων : 18**

πινακασ περιεχομενων

**ΠΕΡΙΛΗΨΗ3**

**ΕΙΣΑΓΩΓΗ4**

Εισαγωγή στον ερευνητικό χώρο της εργασίας4

Βασικές προσεγγίσεις επίλυσής του προβλήματος4

Συνεισφορά της εργασίας 5

**ΕΡΕΥΝΗΤΙΚΟΣ ΧΩΡΟΣ**5

**ΑΛΓΟΡΙΘΜΟΙ ΜΗΧΑΝΙΚΗΣ ΜΑΘΗΣΗΣ**5

SVM (support Vector machine)5

KNN (k-nearest neighbor)6

Random Forest6

Naive Bayes6

**ΜΕΘΟΔΟΛΟΓΙΚΗ ΔΙΑΔΙΚΑΣΙΑ7**

Περιγραφή δεδομένων7

Περιγραφή προ επεξεργασίας 8

Πειράματα μάθησης 13

Ανάλυση και αξιολόγηση 14

**ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΠΡΟΤΑΣΕΙΣ ΓΙΑ ΜΕΛΛΟΝΤΙΚΕΣ ΒΕΛΙΩΣΕΙΣ**17

**ΒΙΒΛΙΟΓΡΑΦΙΑ18**

# περιλιψη

Η μηχανική μάθηση αποτελεί μεγάλο κομμάτι της επιστήμης των υπολογιστών, όσο και της καθημερινότητάς μας. Ο τρόπος με τον οποίο ένα ηλεκτρονικός υπολογιστής έχει την ικανότητα να προβλέψει αποτελέσματα για προβλήματα του πραγματικού κόσμου, είναι με τη χρήση αλγορίθμων μηχανικής μάθησης. Στην εργασία αυτή θα αναλύσουμε ένα σετ δεδομένων, θα εφαρμόσουμε σε αυτό διάφορες μεθόδους προ επεξεργασίας και θα εκπαιδεύσουμε τέσσερεις αλγορίθμους ταξινόμησης. Στο τέλος θα συγκρίνουμε τις επιδώσεις τους για το ίδιο σετ δεδομένων, καθώς επίσης και το πόσο σωστά η όχι ταξινομήσανε τα δεδομένα.

# εισαγωγη

##### Εισαγωγη στον ερευνητικο χωρο της εργασιας

Η μηχανική μάθηση είναι ένα κομμάτι της επιστήμης των υπολογιστών, το οποίο είναι σχετικά καινούργιο στον κλάδο αυτό. Με τον όρο data mining εννοείται η εξόρυξη δεδομένων, δηλαδή η ανακάλυψη γνώσης και συσχετισμών από ένα σετ δεδομένων (data set).

Σκοπός είναι να συλλέξουμε πληροφορίες ή πρότυπα από τα δεδομένα, με τη χρήση αλγορίθμων ομαδοποίησης, κατηγοριοποίησης, στατιστικής, τεχνητής νοημοσύνης και μηχανικής μάθησης. Βασική επιδίωξη της εξόρυξης δεδομένων είναι η πληροφορία που θα εξαχθεί, καθώς και τα πρότυπα που θα προκύψουν να είναι κατανοητά προς τον άνθρωπο. Η απαίτηση αυτή είναι απαραίτητη, προκειμένου ο αλγόριθμος να εξάγει όσο πιο σωστά αποτελέσματα γίνεται.

Μερικές από τις βασικότερες διεργασίες στη εξόρυξη δεδομένων είναι η:

* Ταξινόμηση (Classification)
* Ομαδοποίηση / συσταδοποίηση (Clustering)
* Εύρεση κανόνων συσχέτισης (Association Rule Discovery)
* Εύρεση διαδοχικών / σειριακών προτύπων (Sequential Pattern Discovery)
* Παλινδρόμηση (Regression)
* Ανίχνευση Αποκλίσεων (Deviation Detection)

##### Βασικες προσεγγισεις επιλυσης του προβλήματος

Το αντικείμενο που πραγματεύεται η εργασία αυτή έχει να κάνει συγκεκριμένα με αλγορίθμους ταξινόμησης, καθώς και με την σύγκριση των αποδόσεών τους. Για την υλοποίηση του σκοπού αυτού χρησιμοποιήθηκαν τέσσερεις αλγόριθμοι ταξινόμησης με το ίδιο σετ δεδομένων.

Το σετ δεδομένων (data set) που επιλέχθηκε περιέχει δεδομένα πελατών από διάφορες τράπεζες. Ο βασικός σκοπός ήταν η ταξινόμηση των πελατών με βάση την παραμονή τους ή όχι στην εκάστοτε τράπεζα.

Χρησιμοποιήθηκαν διάφορα εργαλεία για την επίτευξη του σκοπού αυτού, όπως το jupyter notebook και διάφορες βιβλιοθήκες της γλώσσας python.

Για την επίτευξη του σκοπού αυτού ακολουθήθηκαν οι παρακάτω μεθοδολογίες:

1. Εισαγωγή του σετ δεδομένων (.csv)
2. Ανάλυση του σετ δεδομένων και κατανόηση των χαρακτηριστικών τους
3. Προ επεξεργασία των δεδομένων
4. Προσαρμογή των δεδομένων για τους αλγορίθμους
5. Εκτέλεση των αλγορίθμων ταξινόμησης με το σετ δεδομένων
6. Σύγκριση των αποτελεσμάτων του καθενός

##### Συνεισφορα της εργασιας

Με την εργασία αυτή θα μελετήσουμε τέσσερεις αλγόριθμους ταξινόμησης. Εκτελώντας τις τεχνικές μάθησης που προσφέρονται από τη βιβλιοθήκη sklearn της γλώσσας προγραμματισμού python, θα μελετήσουμε την αποδοτικότητά τους. Σκοπός μας είναι να βρούμε ποιος από τους τέσσερεις αλγορίθμους, προσφέρει τη μεγαλύτερη απόδοση και παράλληλα να εντοπίσουμε σε ποια σημεία υστερούν οι άλλοι.

Ο τρόπος με τον οποίο θα καταλήξουμε στον αλγόριθμο με τη μεγαλύτερη απόδοση είναι με τη χρήση της αποδοτικότητας αλλά και με τη χρήση πινάκων σύγχυσης (confusion matrix) για την εύρεση των λανθασμένων ταξινομήσεων.

# Αλγόριθμοι μηχανικής μάθησης

Στο σημείο αυτό θα αναλυθούν οι αλγόριθμοι ταξινόμησης που χρησιμοποιήθηκαν για τη διεξαγωγή των αποτελεσμάτων. Κάθε αλγόριθμος που επιλέχθηκε εκτελέσθηκε με το ίδιο σετ δεδομένων προκειμένου να γίνει αντικειμενική μελέτη.

##### SVM (support Vector machine) [1]

Στη μηχανική µάθηση, τα support vector machines (SVMs) εποπτεύονται μοντέλα µάθησης µε συνδεδεμένους αλγορίθμους µάθησης οι οποίοι αναλύουν δεδομένα και αναγνωρίζουν πρότυπα, που χρησιμοποιούνται για την ταξινόμηση και την ανάλυση παλινδρόμησης.

Η βασική SVM παίρνει ένα σύνολο δεδομένων εισόδου και προβλέπει, για κάθε δεδομένη είσοδο, ποια από τις δύο πιθανές καταστάσεις αποτελεί την έξοδο, καθιστώντας το ένα µηγραµµικό δυαδικό ταξινοµητή. Λαμβάνοντας υπόψη ένα σύνολο παραδειγμάτων εκπαίδευσης, ένας αλγόριθμος SVM χτίζει ένα µοντέλο που εκχωρεί νέα δείγματα σε µία από τις καταστάσεις.

Ένα µοντέλο SVM είναι µια αναπαράσταση των δειγμάτων ως σημεία στο χώρο, τα οποία χαρτογραφούνται έτσι ώστε τα δείγματα διαφορετικών κατηγοριών να µπορούν να διαχωρίζονται όσο το δυνατόν καλύτερα. Εκτός από την εκτέλεση γραµµικής ταξινόμησης, τα SVMs µπορούν να εκτελέσουν το ίδιο αποτελεσματικά και την µη-γραµµική ταξινόμηση.

##### KNN (k-nearest neighbor) [2]

Ο αλγόριθμος k-nearest-neighbors είναι μια μη παραμετρική μέθοδος που χρησιμοποιείται για ταξινόμηση. Οι είσοδοι αποτελούνται από τα k κοντινότερα παραδείγματα στο χώρο. Η έξοδος, ωστόσο, εξαρτάται από την χρήση του k-NN, αν γίνεται για ταξινόμηση.

Στην ταξινόμηση, η έξοδος του αλγόριθμου είναι μέλος μιας κατηγορίας. Ένα αντικείμενο κατηγοριοποιείται από την πλειοψηφία των ψήφων των γειτόνων του, με το αντικείμενο να ανατίθεται στην κλάση που είναι περισσότερο κοινή στους k κοντινότερους γείτονες (με το k να είναι ένας θετικός ακέραιος, συνήθως μικρός αριθμός). Αν k = 1, τότε το αντικείμενο ανατίθεται, απλά, στην κλάση του πιο κοντινού γείτονα.

##### Random Forest [3]

Τα random forests ή random decision forests είναι μια συνδυαστική μέθοδος μάθησης για ταξινόμηση, τα οποία λειτουργούν με την δημιουργία ενός πλήθους decision trees. Κατά την φάση της εκπαίδευσης παράγουν την κλάση που εμφανίζεται πιο συχνά στην ταξινόμηση του κάθε δέντρου. Τα random forests αντιμετωπίζουν και διορθώνουν σε μεγάλο βαθμό το πρόβλημα των decision trees κατά το οποίο υπάρχει overfitting του σετ εκπαίδευσης.

##### Naive Bayes [4]

Ο Naive Bayes ταξινομητής προσφέρει μια απλή προσέγγιση στα προβλήματα, όπου στόχος μας είναι να προβλέψουμε επακριβώς την κατηγορία-κλάση των στιγμιότυπων δοκιμής. Για την επίτευξη της πρόβλεψης χρησιμοποιεί ταξινομημένα στιγμιότυπα εκπαίδευσης που περιλαμβάνουν την πληροφορία της κλάσης που ανήκουν. Βασίζεται σε δυο σημαντικές υποθέσεις.

* Ότι κάθε χαρακτηριστικό των στιγμιότυπων είναι στοχαστικά ανεξάρτητα των υπολοίπων, δεδομένης της κλάσης
* Ότι δεν υπάρχουν άλλα κρυφά χαρακτηριστικά που να επηρεάζουν την διαδικασία της πρόβλεψης.

# Μεθοδολογικη διαδικασια

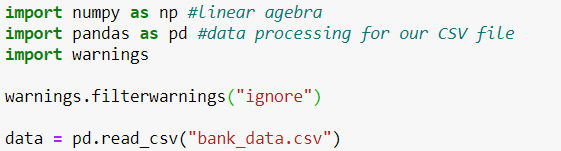
Σε αυτό το κεφάλαιο θα αναλυθεί το σετ δεδομένων που χρησιμοποιήθηκαν. Θα αναλυθούν δηλαδή τα χαρακτηριστικά που το αποτελούν, ποιες είναι οι συσχετίσεις που δημιουργούνται μεταξύ τους, καθώς και η προ επεξεργασία που εφαρμόστηκε. Επιπλέον θα αναλυθούν και οι αλγόριθμοι ταξινόμησης μαζί με την εφαρμογή τους στο σετ δεδομένων. Τέλος θα παρουσιασθούν τα αποτελέσματα που προέκυψαν.

##### Περιγραφη δεδομενων

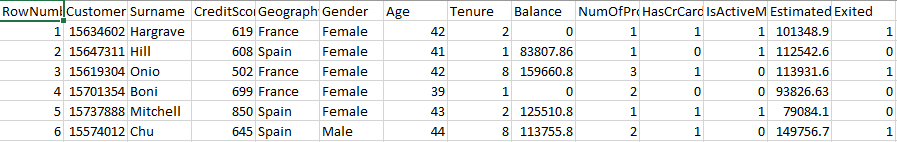
Όπως αναφέρθηκε και στην εισαγωγή, το σετ δεδομένων που χρησιμοποιήθηκε περιέχει στοιχεία πελατών από διάφορες τράπεζες. Στόχος είναι να εντοπίσουμε εάν ένας πελάτης αποχωρεί από την τράπεζα (κλείνοντας το λογαριασμό του), ή εάν συνεχίζει να χρησιμοποιεί τις υπηρεσίες της.

Το data set μας είναι ένα αρχείο τύπου (.csv), οπότε μπορούμε πολύ εύκολα να το επεξεργαστούμε με τη χρήση της βιβλιοθήκης pandas της python. Περιέχει 10.000 γραμμές ,από τις οποίες η κάθε μία περιέχει 14 χαρακτηριστικά που αντιστοιχούν σε κάθε πελάτη από μία τράπεζα.

Η είσοδος των δεδομένων γίνεται με την παρακάτω μέθοδο



Πρώτού προβούμε στην προ επεξεργασία, θα χρειαστεί πρώτα να κατανοήσουμε τα δεδομένα του data set



Το δεδομένο που θέλουμε να εξετάσουμε ονομάζεται exited αποτελεί ένα binary variable (δυαδική μεταβλητή). Πρόκειται δηλαδή για μία μεταβλητή της οποίας οι τιμές είναι αληθείς (true) ή ψευδείς (false). Αλλιώς κυμαίνονται από τις τιμές 0 – 1.

##### Περιγραφη προεπεξεργασιας

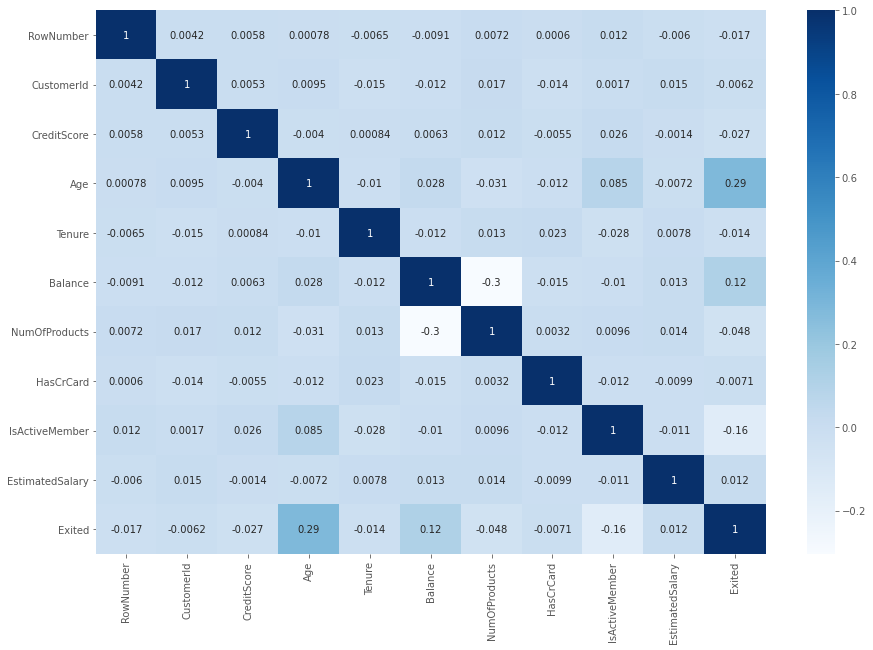
Προκειμένου οι αλγόριθμοι να έχουν μεγαλύτερη αποδοτικότητα, θα πρέπει το data set να υποστεί κάποιες επεξεργασίες. Η διαδικασία αυτή ονομάζεται προ επεξεργασία (preprocessing) και εφαρμόζεται πριν τη χρήση κάποιου αλγόριθμου. Στη συνέχεια θα πρέπει να διαχωρίσουμε το data set σε Ζητούμενα (y) και δεδομένα (x).

###### Δημιουργια heat-map δεδομενων

Προτού προχωρήσουμε στις μεθόδους προ επεξεργασίας, μία καλή τεχνική για να εντοπίσουμε διάφορες συσχετίσεις στα δεδομένα είναι να δημιουργήσουμε ένα heatmap. Πρόκειται για έναν πίνακα ο οποίος οπτικοποιεί το πόσο ένα δεδομένο συσχετίζεται με ένα άλλο.

Για τη δημιουργία του χρησιμοποιήθηκαν οι βιβλιοθήκες της python matplotlib και seaborn. Τα χαρακτηριστικά που διακρίνονται στον παρακάτω πίνακα είναι τα ακόλουθα:

* Το heatmap εμφανίζει μερικές χρήσιμες συσχετίσεις, όπως για παράδειγμα η στήλη 'Age' συσχετίζεται θετικά με τη στήλη (positive related) 'Exited'
* Τα στοιχεία των οποίων το χρώμα είναι πιο σκούρο συσχετίζονται περισσότερο μεταξύ του
* Αντίθετα τα στοιχεία με πιο ανοιχτό χρώμα δε συσχετίζονται τόσο πολύ. Για παράδειγμα η στήλη 'NumOfProducts' δε συσχετίζεται με τη στήλη 'Balance'



###### Αχρείαστες στήλες και διαγραφή

Για την επίλυση του προβλήματος, θα διαγράψουμε κάποια χαρακτηριστικά που δε χρειάζονται. Αυτά είναι τα παρακάτω:

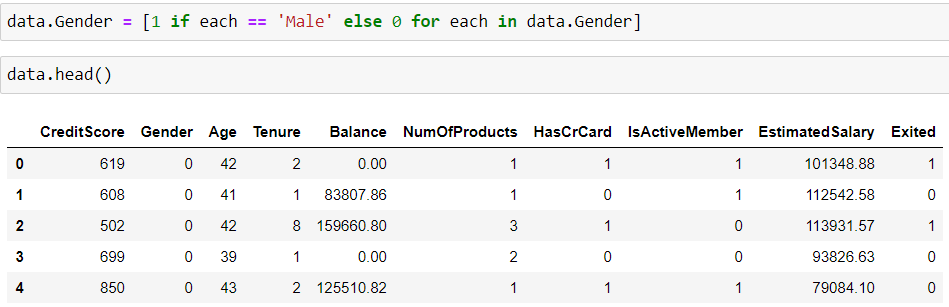
* RowNumber
* CustomerId
* Surname
* Geography

Ο λόγος για τον οποίο τα διαγράφουμε, είναι διότι περιέχουν πληροφορίες οι οποίες δε συσχετίζονται με το ζητούμενό μας που είναι το στοιχείο Exited.

**axis** = 1: επιλογή του άξονα με τις στήλες **inplace** = True: Διαγραφή χωρίς εκ νέου ανάθεσης

###### Αλλαγή των τιμών σε 0 και 1

Στο data set το χαρακτηριστικό του γένους περιέχει αλφαριθμητικά. Για να το απλοποιήσουμε αυτό θα μετατρέψουμε τα Female/Male σε 0/1



Όπως διακρίνουμε τα δεόμενα στη στήλη Gender έχουν γίνει 0.  
Άρα αυτό σημαίνει ότι οι πέντε πρώτες γραμμές περιέχουν πελάτες γένους θηλυκού.

###### Διαχωρισμός δεδομένων

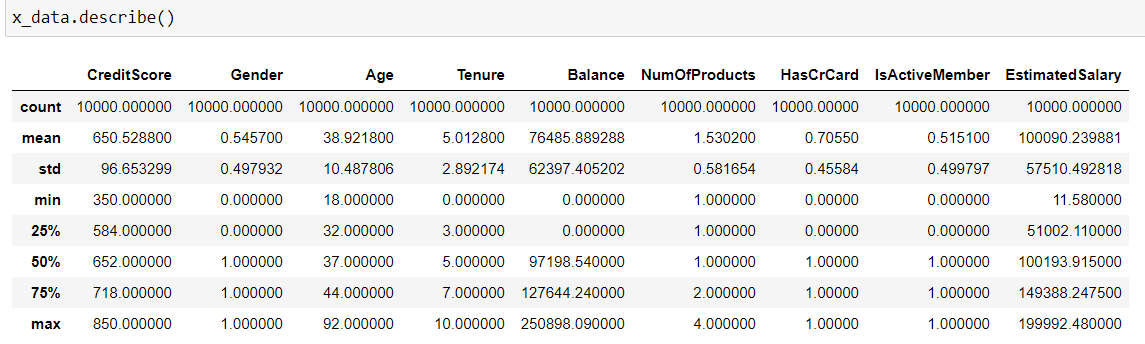
Σε αυτό το σημείο θα διαχωρίσουμε το ζητούμενό μας από το data set, έτσι ώστε να δημιουργήσουμε το τις μεταβλητές x\_data και y όπου περιέχουν τα δεδομένα και τις απαντήσεις αντίστοιχα

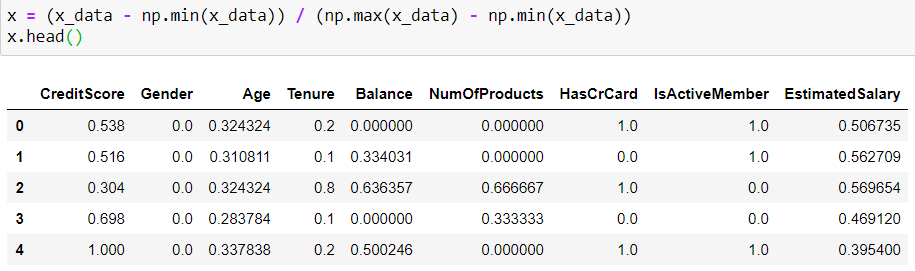
Επιπλέον θα ήταν καλό να οπτικοποιήσουμε τον πίνακα y έτσι ώστε να έχουμε καλύτερη άποψη πάνω γύρω από το ζητούμενο. Όπως και με το hetamap, θα οπτικοποιήσουμε τον πίνακα με τη χρήση των βιβλιοθηκών matplotlib και seaborn.

###### Κανονικοποiηση (Normalization)

Εάν εμφανίσουμε τον πίνακα x\_data θα με τη μέθοδο describe() της βιβλιοθήκης pandas, εντοπίζουμε πως υπάρχουν αριθμοί μεγάλης κλίμακας στο x\_data.



Θα πρέπει να κανονικοποιήσουμε τις τιμές αυτές σε 0 και 1. Η κανονικοποίηση που εφαρμόστηκε στα δεδομένα έγινε με τον παρακάτω τύπο:

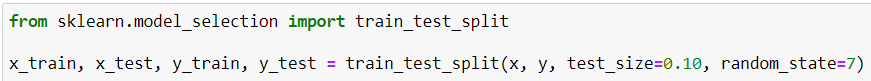


Όπως βλέπουμε οι τιμές έχουν συμπιεστεί σε τιμές μεταξύ του 0 και του 1. Αυτό το κάναμε προκειμένου η υπολογιστική ισχύς που θα χρειαστεί για την ανάλυση του από τους αλγορίθμους να είναι μικρότερη.

###### ΠροσαρμογΗ των δεδομένων για εκπαδευση

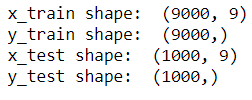
Σε αυτό το σημείο Θα διαιρέσουμε τα δεδομένα σε (x\_train y\_train) και σε (x\_test y\_test). Στη συνέχεια θα εκπαιδεύουμε τους αλγορίθμους με τα δεδομένα αυτά και θα προσπαθήσουν να προβλέψουν τα test δεδομένα. Τέλος θα συγκρίνουμε τις προβλέψεις (y\_pred) με τα test δεδομένα.

Ο τρόπος με τον οποίο θα χωρίσουμε τα δεδομένα είναι με την παρακάτω συνάρτηση.



* x, y: Το data set
* test\_size = 0.10: Μέγεθος των test μεταβλητών
* random\_state = 7: Τυχαίος αριθμός ως seed

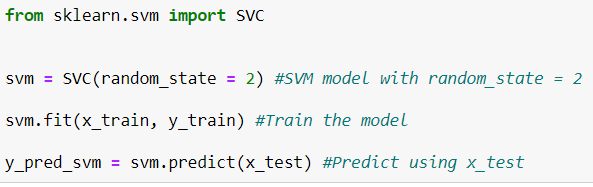
Με τον τρόπο αυτό χωρίσαμε τα δεδομένα μας όπως απεικονίζονται παρακάτω.



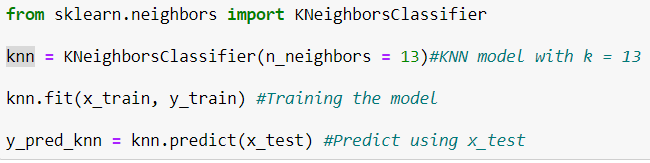
##### Πειραματα μαθησης

Στην ενότητα αυτή θα αναλύσουμε την εκτέλεση του κάθε αλγορίθμου, καθώς και την κλήση τους από τη βιβλιοθήκη sklearn της python. Στις παρακάτω εικόνες εμφανίζονται τόσο τα ορίσματα που δέχεται ο κάθε αλγόριθμος, όσο και οι τρόποι με τους οποίους το μοντέλο εκπαιδεύεται και στη συνέχεια προβλέπει. Η εκπαίδευση γίνεται με τα δεδομένα που περιέχουν το x\_train και y\_train, ενώ η πρόβλεψη γίνεται στα δεδομένα του x\_test.

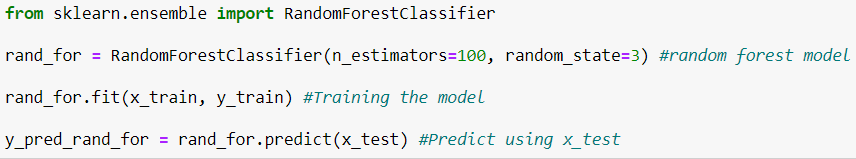
###### SUPPORT VECTOR MACHINE (SVM) Αλγόριθμος



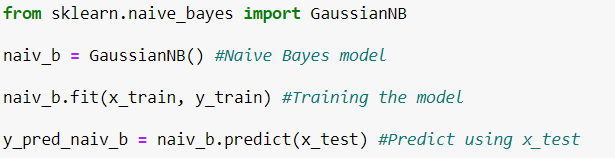
###### K-NEAREST NEIGHBOR (KNN) Αλγοριθμος



###### RANDOM FOREST Αλγοριθμος



###### NAIVE BAYES Αλοριθμος

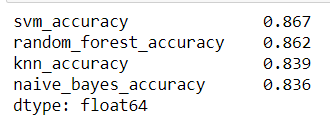


##### Αναλυση και αξιολογηση

Τέλος θα συγκρίνουμε τα μοντέλα που εκπαιδεύσαμε. Για το κάθε μοντέλο με την ολοκλήρωση της εκπαίδευσής του, εξάγομε το ποσοστό ακρίβειας που είχε στο συγκεκριμένο σετ δεδομένων. Επιπλέον δημιουργήσαμε έναν confusion matrix για τον κάθε αλγόριθμο, ώστε να παρατηρήσουμε σε ποια σημεία έκανε λάθος ταξινόμηση.

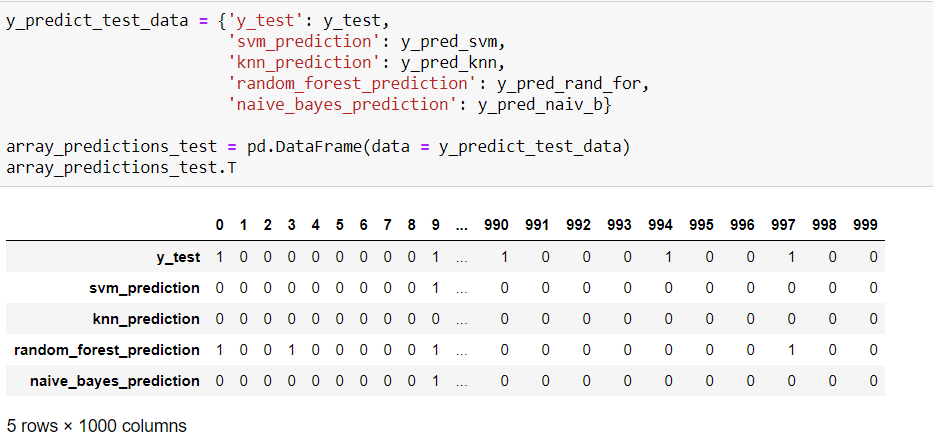
###### Συγκριση των αποδοσεων

Στον παρακάτω πίνακα εμφανίζονται οι αποδώσεις που είχαν οι αλγόριθμοι:

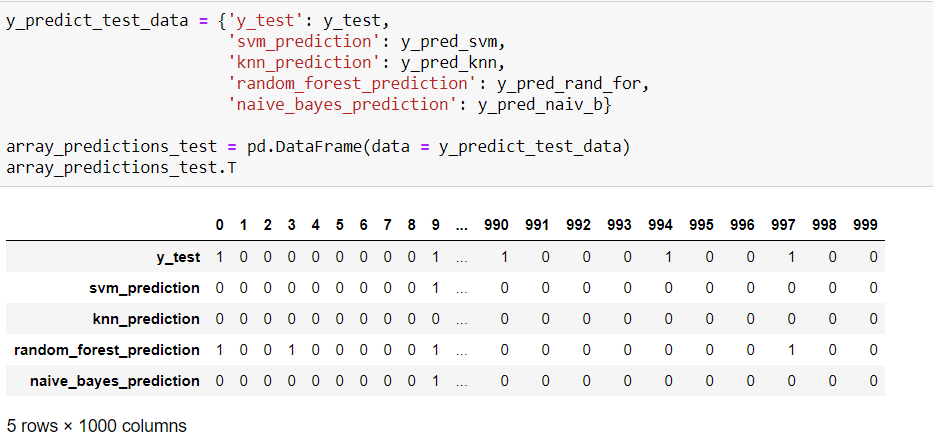


Όπως μπορούμε να δούμε και παραπάνω ο αλγόριθμος SVM προβλέπει με απόδοση 0.87

Μία ακόμη μέθοδος απεικόνισης των αποτελεσμάτων που πραγματοποιήθηκε είναι η σύγκριση των αποτελεσμάτων που προέβλεψε ο κάθε αλγόριθμος, με τα πραγματικά αποτελέσματα του πίνακα y\_test, για κάθε στοιχείο του πίνακα αυτού.



Με άλλα λόγια θα κατασκευάσουμε ένα DataFrame με τη χρήση της βιβλιοθήκης pandas, για την καλύτερη προβολή των αποτελεσμάτων.

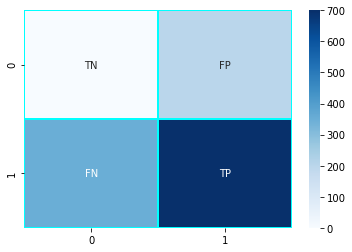


Ένα παράδειγμα με το οποίο μπορούμε να διατυπώσουμε τον παραπάνω πίνακα είναι το εξής. Για το στοιχείο μηδέν η πραγματική τιμή του y\_test είναι 1, δηλαδή ο πελάτης θα αποχωρήσει από την συγκεκριμένη τράπεζα. Για το ίδιο στοιχείο, όλοι οι αλγόριθμοι επέστρεψαν την τιμή μηδέν, εκτός από τον random forest, πράγμα που τον καθιστά τον μόνο σωστό για το στοιχείο αυτό.

###### συγκριση των confusion matrixes

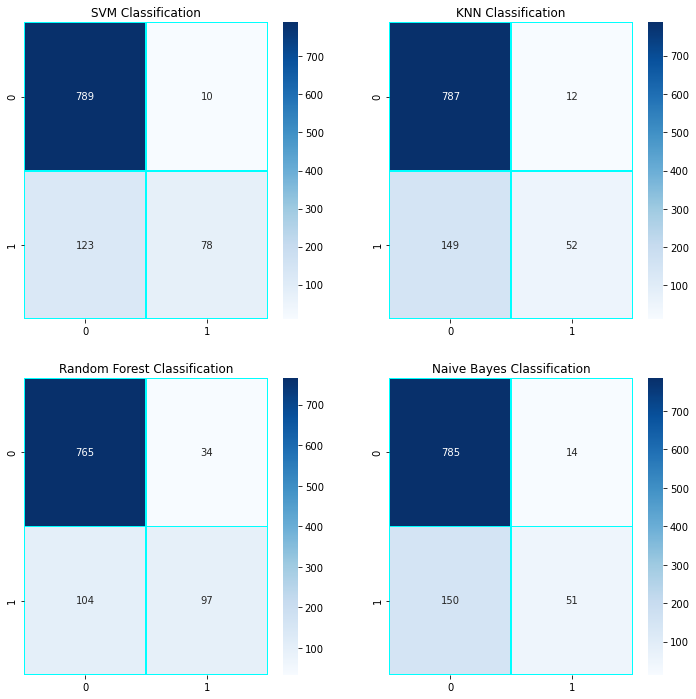
Τέλος για την κατανόηση του ποιος αλγόριθμος κατάφερε να προβλέψει σωστά ή όχι τις τιμές του y\_test, συγκρίνουμε τους Confusion Matrixes μεταξύ τους. Κάθε πίνακας περιέχει το παρακάτω layout:

Σωστές προβλέψεις

* TN (True Negatives): Τιμές που είναι αρνητικές και ο αλγόριθμος τις ταξινόμησε σωστά
* TP (True Positives): Τιμές που είναι θετικές και ο αλγόριθμος τις ταξινόμησε σωστά

Λανθασμένες προβλέψεις

* FN (False Negative): Τιμές που είναι αρνητικές και ο αλγόριθμος δεν ταξινόμησε σωστά
* FP (False Positive): Τιμές που είναι θετικές και ο αλγόριθμος δεν ταξινόμησε σωστά



Στους παραπάνω πίνακες μπορούμε να διακρίνουμε τις εξής πληροφορίες:

* Αλγόριθμος SVM
  + TN = 789
  + TP = 78
* Αλγόριθμος KNN
  + TN = 787
  + TP = 52
* Αλγόριθμος Random Forest
  + TN = 765
  + TP = 97
* Αλγόριθμος Naïve Bayes
  + TN = 785
  + TP = 51

# Συμπεράσματα και προτάσεις για μελλοντικές βελτιώσεις

Ολοκληρώνοντας την εργασία αυτή μας δόθηκε η ευκαιρία να ασχοληθούμε με τις μεθόδους μηχανικής, όπως επίσης και την προ επεξεργασία των δεδομένων. Έγιναν επίσης αντιληπτοί αρκετοί από τους τρόπους εύρεσης συσχετίσεων στο σετ των δεδομένων, για την καλύτερη προ επεξεργασία αλλά και την κατανόηση των χαρακτηριστικών. Επιπλέον είδαμε τους τρόπους σύγκρισης αλγορίθμων μάθησης και με ποιόν τρόπο θα μπορούσε να εφαρμοστεί σε ένα πραγματικό πρόβλημα.

Μερικές μελλοντικές βελτιώσεις θα ήταν η προσθήκη περισσότερων αλγορίθμων. Επιπλέον θα μπορούσε το απλό πρόγραμμα αυτό να υλοποιηθεί με τέτοιο τρόπο ώστε να δέχεται αρκετά data sets και στη συνέχεια να εμφανίζει στο χρήστη ποιος είναι ο καλύτερος αλγόριθμος για το εκάστοτε data set.

# βιβλιογραφια

1. Suykens, J. A. K., & Vandewalle, J. (1999). Neural Processing Letters, 9(3), 293–300.
2. Dudani, S. A. (1976). The Distance-Weighted k-Nearest-Neighbor Rule. IEEE Transactions on Systems, Man, and Cybernetics, SMC-6(4), 325–327
3. Pal, M. (2005). Random forest classifier for remote sensing classification. International Journal of Remote Sensing, 26(1), 217–222.
4. Lewis, D. D. (1998). Naive (Bayes) at forty: The independence assumption in information retrieval. In Machine Learning: ECML-98 (pp. 4–15).
5. Jupyter Notebook - <https://jupyter.org/>
6. Sklearn - <https://scikit-learn.org/stable/>