**ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ**
**Σχολή Χρηματοοικονομικής και Στατιστικής**

**Τμήμα Στατιστικής και Ασφαλιστικής Επιστήμης**

ΜΕΤΑΠΤΥΧΙΑΚΟ ΠΡΟΓΡΑΜΜΑ ΣΠΟΥΔΩΝ
ΣΤΗΝ **ΕΦΑΡΜΟΣΜΕΝΗ ΣΤΑΤΙΣΤΙΚΗ**

# ΜΕΘΟΔΟΙ ΜΕΛΕΤΗΣ ΤΟΥ ΡΥΘΜΟΥ ΑΠΩΛΕΙΑΣ ΠΕΛΑΤΩΝ ΚΑΙ ΤΗΣ ΑΞΙΑΣ ΣΥΝΟΛΙΚΟΥ ΧΡΟΝΟΥ ΖΩΗΣ ΠΕΛΑΤΗ

Νικόλαος Δ. Βαρελάς

Διπλωματική Εργασία

που υποβλήθηκε στο Τμήμα Στατιστικής και Ασφαλιστικής Επιστήμης του Πανεπιστημίου Πειραιώς ως μέρος των απαιτήσεων για την απόκτηση του Μεταπτυχιακού Διπλώματος Ειδίκευσης στην *Εφαρμοσμένη Στατιστική*

Πειραιάς
Μάρτιος 2019

**Π Α Ν Ε Π Ι Σ Τ Η Μ Ι Ο  Π Ε Ι Ρ Α Ι Ω Σ**

**Σχολή Χρηματοοικονομικής και Στατιστικής**

**Τμήμα Στατιστικής και Ασφαλιστικής Επιστήμης**

ΜΕΤΑΠΤΥΧΙΑΚΟ ΠΡΟΓΡΑΜΜΑ ΣΠΟΥΔΩΝ
ΣΤΗΝ **ΕΦΑΡΜΟΣΜΕΝΗ ΣΤΑΤΙΣΤΙΚΗ**

# ΜΕΘΟΔΟΙ ΜΕΛΕΤΗΣ ΤΟΥ ΡΥΘΜΟΥ ΑΠΩΛΕΙΑΣ ΠΕΛΑΤΩΝ ΚΑΙ ΤΗΣ ΑΞΙΑΣ ΣΥΝΟΛΙΚΟΥ ΧΡΟΝΟΥ ΖΩΗΣ ΠΕΛΑΤΗ

Νικόλαος Δ. Βαρελάς

Διπλωματική Εργασία

που υποβλήθηκε στο Τμήμα Στατιστικής και Ασφαλιστικής Επιστήμης του Πανεπιστημίου Πειραιώς ως μέρος των απαιτήσεων για την απόκτηση του Μεταπτυχιακού Διπλώματος Ειδίκευσης στην *Εφαρμοσμένη Στατιστική*

Πειραιάς
Μάρτιος 2019

Η παρούσα Διπλωματική Εργασία εγκρίθηκε ομόφωνα από την Τριμελή Εξεταστική Επιτροπή που ορίσθηκε από τη ΓΣΕΣ του Τμήματος Στατιστικής και Ασφαλιστικής Επιστήμης του Πανεπιστημίου Πειραιώς στην υπ' αριθμ. ……. συνεδρίασή του σύμφωνα με τον Εσωτερικό Κανονισμό Λειτουργίας του Προγράμματος Μεταπτυχιακών Σπουδών στην Εφαρμοσμένη Στατιστική

Τα μέλη της Επιτροπής ήταν:

- Καθηγητής Κούτρας Μάρκος (Επιβλέπων)
- Καθηγητής Θεοδωρίδης Ιωάννης
- Καθηγητής Τσίμπος Κλέων

Η έγκριση της Διπλωματική Εργασίας από το Τμήμα Στατιστικής και Ασφαλιστικής Επιστήμης του Πανεπιστημίου Πειραιώς δεν υποδηλώνει αποδοχή των γνωμών του συγγραφέα.

**UNIVERSITY OF PIRAEUS**

**School of Finance and Statistics**

**Department of Statistics and Insurance Science**

POSTGRADUATE PROGRAM IN
**APPLIED STATISTICS**

# TECHNIQUES FOR STUDYING CHURN AND CUSTOMER LIFETIME VALUE

By

Nikolaos D. Varelas

MSc Dissertation

submitted to the Department of Statistics and Insurance Science of the University of Piraeus in partial fulfilment of the requirements for the degree of Master of Science in Applied Statistics

Piraeus, Greece
March 2019

*Στην Οικογένεια μου*

# Ευχαριστίες

Σε αυτό το σημείο θα ήθελα να ευχαριστήσω τον επιβλέποντα μου κ. Μάρκο Κούτρα, καθηγητή του τμήματος Στατιστικής και Ασφαλιστικής επιστήμης του Πανεπιστημίου Πειραιώς, για την ανάθεση του θέματος και την συνεχή καθοδήγηση που μου παρείχε καθ' όλη τη διάρκεια της συγγραφής της διπλωματικής εργασίας ώστε να λάβει την παρούσα μορφή της. Ευχαριστώ επίσης τα μέλη της τριμελούς επιτροπής, τον Καθηγητή κ. Κ. Τσίμπο και τον Καθηγητή κ. Ι. Θεοδωρίδη για τον χρόνο που αφιέρωσαν στην διόρθωση της εργασίας. Επιπλέον ευχαριστώ θερμά την οικογένεια μου για την υποστήριξη που μου έχει προσφέρει. Τους γονείς μου για την οικονομική και ψυχολογική υποστήριξη και τον αδερφό μου για τις πολύτιμες συμβουλές και γνώσεις που μου παρείχε καθ'όλη τη διάρκεια των σπουδών μου. Τέλος θα ήθελα να ευχαριστήσω την κοπέλα μου που σε αυτή την προσπάθεια ήταν συνέχεια δίπλα μου και με στήριζε.

# Περίληψη

Η παρούσα διπλωματική έχει ως θέμα τη πρόβλεψη του ρυθμού απώλειας και την συνολική αξία πελάτη. Τα τελευταία χρόνια πολλές επιχειρήσεις έχουν στη διάθεση τους τεράστιο όγκο δεδομένων. Τα δεδομένα αυτά μπορούν να χρησιμοποιηθούν για την δημιουργία προβλέψεων για τον ρυθμό απώλειας και συνολική αξίας πελάτη. Ωστόσο οι κλασσικές μέθοδοι που χρησιμοποιούνται μέχρι σήμερα αδυνατούν να εκμεταλευτούν τον μεγάλο όγκο δεδομένων που είναι διάθεσιμος σήμερα.

Στόχος της εργασίας είναι αρχικά η θεωρητική περιγραφή των σύγχρονων μοντέλων μηχανικής μάθησης και έπειτα η εφαρμογή των μοντέλων σε πραγματικά δεδομένα. Συγκεκριμένα στο πρώτο κεφάλαιο γίνεται μία αναφορά στην σημασία της πρόβλεψης του ρυθμού απώλειας, συνολικής αξίας πελάτη και κλασσικές τεχνικές που χρησιμοποιούνται σήμερα. Στα επόμενα τρία γίνεται περιγραφή μοντέλων κατηγοριοποίησης, παλινδρόμησης και τεχνικές μείωσης διαστάσεων. Στο πέμπτο κεφάλαιο κατασκευάζονται μεταβλητές με σκοπό την πρόβλεψη ρυθμού απώλειας, συνολικής αξίας πελάτη, γίνεται επιλογή των σημαντικότερων και εφαρμόζονται όλες οι τεχνικές που περιγράφηκαν σε προηγούμενες ενότητες. Η σύγκριση των τεχνικών για τον ρυθμό απώλειας έγινε χρησιμοποιώντας το κριτήριο AUC και για την συνολική αξία πελάτη το κριτήριο RMSE. Το υπόδειγμα της Λογιστικής Παλινδρόμησης απέφερε τα καλύτερα αποτελέσματα για την πρόβλεψη του ρυθμού απώλειας πελάτη. Το υπόδειγμα LightGBM είχε καλύτερη προσαρμογή από ότι η τεχνική Random Forest το οποίο επιβεβαιώνει την χρησιμότητας της.

# Abstract

The subject of the present MSc Dissertation is the prediction of churn's rate and customer's life time value. In recent years many companies have at their disposal large amount of data. Those data can be used in order to make predictions about the churn's rate and customer's life time value. However, traditional methods that have been used until today cannot exploit the large volume of data been available today.

The aim of this thesis is firstly a theoretically description of the modern models machine learning and then apply these models on a real dataset. In particular, in the first chapter there is a mention in the importance of predicting churn rate, customer life time value and classic techniques used today. In the next three there is a description of classification , regression and dimensional reduction techniques. In the fifth chapter we construct variables to predict customer's churn, life time value, select the most important ones and apply all the techniques described in previous sections. The comparison of customer's churn rate was done using the AUC criterion and for the customer life time value the RMSE criterion. The logistic regression model produced the best results for predicting customer's churn rate. The LightGBM model was better suited than the Random Forest technique that confirms its usefulness.

# Περιεχόμενα

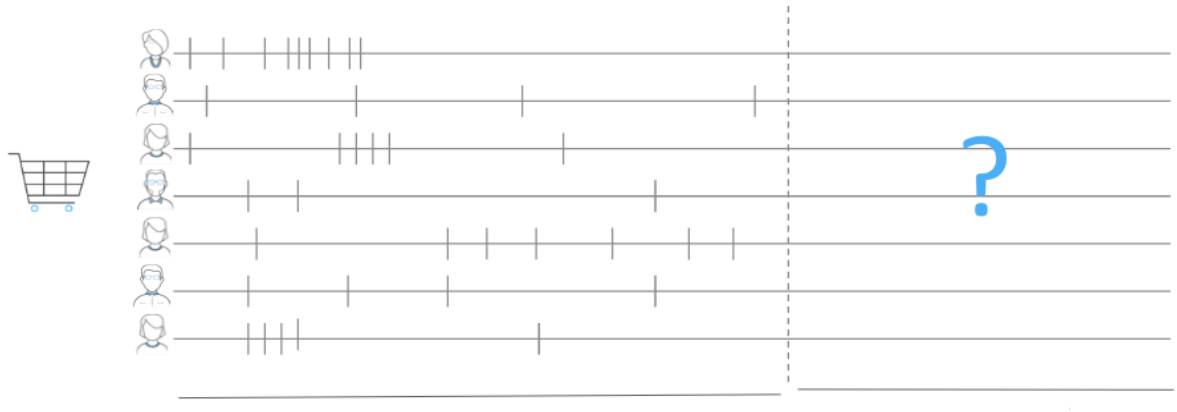
# Κατάλογος Σχημάτων

# Κατάλογος Πινάκων

# 1. Εισαγωγή

Η εκτίμηση της αξίας του χρόνου ζωής των πελατών (Customer Lifetime Value, CLV) παίζει καθοριστικό ρόλο στο σχεδιασμό των στρατηγικών ανάπτυξης και μάρκετινγκ μίας επιχείρησης. Πιο συγκεκριμένα, πρόκειται για μία μέτρηση που καθορίζει την αξία κάθε πελάτη της σε συγκεκριμένο χρονικό διάστημα. Έχοντας εικόνα αυτής, μπορεί μία επιχείρηση να εξασφαλίσει σημαντικά κέρδη με την απόκτηση ενός νέου πελάτη ή τη διατήρηση ενός κερδοφόρου, καθώς και να ελαχιστοποίησει τη ζημιά της με την απόρριψη ενός μη κερδοφόρου.Η συνεισφορά αυτή της CLV στη μελλοντική λειτουργία μίας επιχείρησης προϋποθέτει την ακριβή πρόβλεψή της.

Υπάρχουν πολλές τεχνικές υπολογισμού της CLV. Οι δύο πιο διαδεδομένες είναι οι ιστορικές και οι προβλεπτικές. Οι ιστορικές προσπαθούν να προσδιορίσουν την αξία κάθε πελάτη μόνο βάσει παλαιών συναλλαγών του χωρίς να λαμβάνουν καθόλου υπόψη τη συμπεριφορά του. Η συγκεκριμένη τεχνική είναι αποδεκτή εάν όλοι οι πελάτες συμπεριφέρονται κατά τον ίδιο τρόπο και αλληλεπιδρούν με την επιχείρηση το ίδιο χρονικό διάστημα. Ωστόσο στην πράξη συνήθως δεν ισχύει κάτι τέτοιο. Ας παρατηρήσουμε το γράφημα του σχήματος 1.1. όπου παρατίθενται οι συναλλαγές επτά πελατών μίας επιχείρησης.




Σχήμα 1.1 Συναλλαγές 7 πελατών

Αν θέλαμε να ακολουθήσουμε μία κλασσική ιστορική προσέγγιση, θα παίρναμε το μέσο όρο των παλαιών συναλλαγών τους, ώστε βάσει αυτών να καταλήξούμε στους πολυτιμότερους πελάτες μελλοντικά. Ωστόσο η συγκεκριμένη τεχνική δεν θα ήταν αποδοτική. Εξετάζοντας για παράδειγμα τους δύο πρώτους πελάτες, μπορούμε να υποθέσουμε πως ο πρώτος, αν και έχει περισσότερες συναλλαγές από το δεύτερο, είναι πιθανότερο να μη διατηρήσει σχέσεις με την επιχείρηση. Σε αντίθεση με την ιστορική προσέγγιση, οι προβλεπτικές μέθοδοι λαμβάνουν υπόψη την καταναλωτική συμπεριφορά των πελατών και ως εκ τούτου καταλήγουν σε πιο ακριβή συμπεράσματα για τη CLV κάνοντας χρήση διάφορων στατιστικών τεχνικών.

Οι πρώτες χρονικά στατιστικές τεχνικές μοντελοποίησης της CLV είναι γνωστές ως "Buy Till You Die"(BTYD).Τα συγκεκριμένα μοντέλα χρησιμοποιούν παραμετρικές κατανομές γιά την μοντελοποίηση της CLV και την καταναλωτική συχνότητα του πελάτη. Το πρώτο, που είναι και ένα από τα πιο γνωστά μοντέλα, είναι το Pareto/Negative Binomial το οποίο υποθέτει ότι το πλήθος των συνναλαγών ακολουθεί Αρνητική Διωνυμική. Η πιθανότητα κάποιος πελάτης να σταματήσει να αλλιλεπιδρά με την επιχείρηση(churn) ακολουθεί κατανομή Pareto.

Το πρόβλημα με το συγκεκριμένο μοντέλο είναι ότι δεν λαμβάνει υπόψη καθόλου την αξία της κάθε συναλλαγής. Μία επέκταση των BTYD μοντέλων είναι τα Recency-Frequency Value(RFM) μοντέλα τα οποία συμπεριλαμβάνουν τη χρονική στιγμή που πραγματοποήθηκε η τελευταία αγορά (recency), τον αριθμό των αγορών (frequency) και την αξία των αγορών(monetary) του κάθε πελάτη ώστε να μοντελοποιηθεί η CLV. Για την αξία των αγορών συνήθως υποθέτουυμε ότι ακολουθεί την κατανομή Γάμμα.

Παρ' όλη την επιτυχία που είχαν τα μοντέλα αυτά τα παλαιότερα χρόνια, σήμερα που ο όγκος των δεδομένων μεγαλώνει συνεχώς και τα προβλήματα που αντιμετωπίζει κάποιος αποτελούνται από σημαντικό αριθμό μεταβλητών, οι συγκεκριμένες τεχνικές δεν είναι αποδοτικές. Η χρήση τεχνικών μηχανικής μάθησης θεωρείται αναγκαία.

Στα πλαίσια της παρούσας εργασίας παρουσιάζονται σύγχρονες τεχνικές μοντελοποίησης της CLV βασισμένες σε Δέντρα Παλινδρόμησης και σε Δέντρα Κατηγοριοποίησης για την πρόβλεψη του ρυθμού απώλειας πελατών(Churn rate). Επίσης γίνεται αναφορά στις δύο βασικές κατηγορίες τους, Bagging και Boosting Trees και παρουσιάζονται ορισμένα βασικά μοντέλα με τις σύγχρονες επεκτάσεις τους.

# 2. Μέθοδοι Βασισμένες σε Δέντρα

## 2.1 Δέντρα Παλινδρόμησης

Τα Δέντρα Παλινδρόμησης είναι μία απλή τεχνική, εύκολη στην ερμηνεία η οποία πολλές φορές προσαρμόζεται ικανοποιητικά στα δεδομένα.

Θεωρούμε ένα σύνολο της μορφής $(x_i, y_i)$ για $i = 1,2,..N$ με $x_i = (x_{i1}, x_{i2}, \dots, x_{ip})$ και για $j = 1,2,..,p$.

Η κατασκευή ενός Δέντρου Παλιδρόμησης ορίζεται ως εξής:

- Χωρίζεται το σύνολο τιμών της μεταβλητή στόχος σε Μ περιοχές$R_1, R_2, \dots, R_M$
- Μοντελοποιείται η μεταβλητή σαν σταθερά $c_m$σε κάθε περιοχή.

$$f(x) = \sum_{m=1}^{M} c_m \, I(x \in R_m)$$

Έχοντας σαν κριτήριο ελαχιστοποίησης το άθροιμα των τετραγώνων $\sum(y_i - f(x_i))^2$ εύκολα υπολογίζει κανείς ότι το βέλτιστο $\hat{c}_m$ είναι ο μέσος όρος των $y_i$ στην περιοχή $m$ :

$$c_m = ave(y_i | x_i \in R_m)$$

Το πρόβλημα που δημιουργείται είναι ότι, χρησιμοποιώντας το άθροισμα των τετραγώνων για να βρούμε τον καλύτερο διαχωρισμό και ψάχνοντας ανάμεσα σε όλες τις δυνατές διαιρέσεις καταλήγουμε σεν έναν αλγόριθμο ο οποίος είναι υπολογιστικά χρονοβόρος. Για το λόγο αυτό συνήθως χρησιμοποιείται μία άλλη προσέγγιση σύμφωνα με την οποία σε κάθε βήμα, χωρίζεται η μεταβλητή στόχος σε δύο περιοχές μέσα από δύο κλαδιά, επιλέγεται μία μεταβλητή $X_j$ και $s$ το σημείο διαχωρισμού που αποφέρει την μεγαλύτερη μείωση στο άθροισμα τετραγώνων. Ουσιαστικά ψάχνουμε μία μεταβλητή $j$ και ένα σημείο $s$ έτσι ώστε να ελαχιστοποιηθεί η εξίσωση :

$$\sum_{x_i \in R_1(j,s)} (y_i - c_1)^2 + \sum_{x_i \in R_2(j,s)} (y_i - c_2)^2$$

όπου, $\mathrm{R}_1(j,s) = \{ X|X\_j \leq s\}$ και $R_2(j,s) = \{X|X\_j > s\}$

Στη συνέχεια επαναλαμβάνεται η διαδικασία για κάθε μία περιοχή που δημιουργήθηκε. Το ερώτημα που προκύπτει είναι πόσο θα μεγαλώσουμε τα δέντρα μας. Ένα μεγάλο δέντρο θα έχει ειδικευτεί πολύ στα δεδομένα μας, με αποτέλεσμα να μην έχει καλή προβλεπτική ικανότητα σε νέα δεδομένα που δεν έχει ξανα δεί. Από την άλλη ένα μικρό δέντρο μπορεί να μην έχει εκπαιδευτεί σωστά οπότε θα αποφέρει μη ικανοποιητικά αποτελέσματα.

Μία αντιμετώπιση του προβλήματος θα ήταν να ορίζαμε ένα ελάχιστο κατώφλι και μόνον εάν η μείωση του αθροίσματος των τετραγώνων που επιτυγχάνεται από τον διαχωρισμό είναι μεγαλύτερη από το κατώφλι, τότε να πραγματοποιείται ο διαχωρισμός.

Η συγκεκριμένη στρατηγική δεν είναι πάντα η βέλτιστη καθώς ένας κακός αρχικός διαχωρισμός μπορεί στην συνέχεια να οδηγήσει σε έναν επόμενο πολύ καλό.

Η στρατηγική που αποδίδει καλύτερα είναι αυτή του κλαδέματος του δέντρου. Η ιδέα είναι να μεγαλώσουμε ένα δέντρο με προκαθορισμένο αριθμό κόμβων και στη συνέχεια να το κλαδέψουμε χρησιμοποιώντας ένα κριτήριο ζημιάς της πολυπλοκότητας του δέντρου.Το κριτήριο αναλύεται στην συνέχεια.

- Αρχικά εκπαιδεύουμε ένα δέντρο έστω $\mathrm{T} \subset T_0$ το οποίο μπορεί να είναι οποιοδήποτε δέντρο που προέκυψε από το κλάδεμα του δέντρου $T_0$.
- Θέτουμε τους τερματικούς κόμβους του $T$ με τον κόμβο $m$ να αντιπροσωπεύει την περιοχή $R_m$.

$$\hat{c}_m = \frac{1}{N_m} \sum_{x_i \in R_m} y_i$$

$$Q_m(T) = \sum_{x_i \in R_m} (y_i - \hat{c}_m)^2$$

$$Ca(\mathrm{T}) = \sum_{m=1}^{|T|} Q_m(T) + \alpha|T|$$

Ουσιαστικά ο πρώτος όρος της συνάρτησης $Ca$ μετράει πόσο καλά προσαρμόζεται το δέντρο στα δεδομένα εκπαίδευσης(μικρές τιμές υποδηλώνουν καλή προσαρμογή) και ο δεύτερος όρος

την πολυπλοκότητα του δέντρου. Η παράμετρος $\alpha \geq 0$ υποδηλώνει το αντιστάθισμα μεταξύ πολυπλοκότητας και καλής προσαρμογής του δέντρου.

Για $\alpha = 0$ το δέντρο που προκύπτει είναι το $T_0$, καθώς δεν προσθέτεται κόστος για κάθε κόμβο που περιλαμβάνει το δέντρο. Όσο η παράμετρος $\alpha$ μεγαλώνει το κόστος πολυπλοκότητας του δέντρου αυξάνεται οπότε καταλήγουμε σε μικρότερα δέντρα τα οποία δεν προσαρμόζονται τόσο καλά στα δεδομένα εκπαίδευσης. 'Οσο μικρότερη είναι η παράμετρος $\alpha$ τόσο μεγαλύτερο είναι το δέντρο που κατασκευάζεται με αποτέλεσμα πολλές φορές να υπερειδικεύεται(overfitting) στα δεδομένα εκπαίδευσης με αποτέλεσμα να μην μπορούν να αποδόσουν σε άλλα σύνολα δεδομένων.

## 2.2 Δέντρα Κατηγοριοποίησης

Η τεχνική είναι παρόμοια με αυτή των Δέντρων Παλινδρόμησης το μόνο που χρειάζεται να αλλάξουμε είναι το κριτήριο διαχωρισμού καθώς η μεταβλητή στόχος παίρνει διακριτές τιμές. Εάν η μεταβλητή στόχος παίρνει τις τιμές $1,2, \ldots, k$ τότε σε ένα κόμβο $m$ που ορίζει μία περιοχή $R_m$ με πλήθος παρατηρήσεων $N_m$ ορίζεται η πιθανότητα :

$$\hat{p}_{mk}(m) = \frac{1}{N_m} \sum_{x_i \in R_m} I(y_i = k)$$

η οποία είναι απλά το ποσοστό των παρατηρήσεων της κλάσης $k$ που βρίσκονται στον κόμβο $m$. Κατηγοριοποιούμε τις παρατηρήσεις στον κόμβο $m$ ανάλογα με το ποιά κατηγορία έχει το μεγαλύτερο ποσοστό, $k(m) = argmax_k \widehat{p_{mk}}$ , την πλειοψηφική κατηγορία.

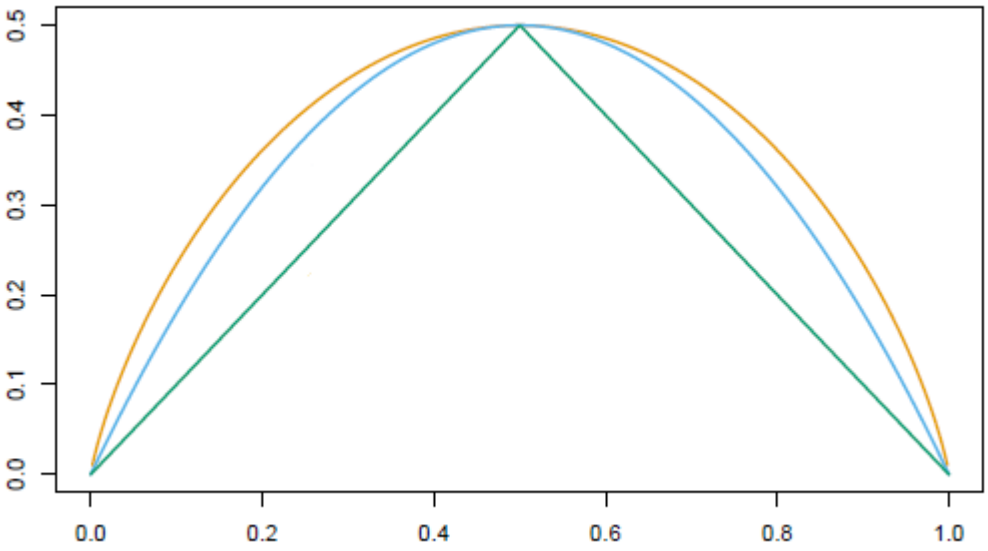
Στα Δέντρα Παλινδρόμησης χρησιμοποιήσαμε το άθροισμα των τετραγώνων ως κριτήριο διαχωρισμού. Στα Δέντρα Κατηγοριοποίησης υπάρχουν τρία κριτήρια διαχωρισμού:

Σφάλμα Λανθασμένης Ταξινόμησης : $\frac{1}{N_m} \sum_{i \in R_m} I\big(y_i \neq k(m)\big) = 1 - \hat{p}_{mk}(m)$

Gini Index : $\sum_{k=1}^{K} \hat{p}_{mk}(1 - \hat{p}_{mk})$

Cross-entropy : $-\sum_{k=1}^{K} \hat{p}_{mk}\, log(\hat{p}_{mk})$

Στην πράξη συνήθως χρησιμοποιούνται τα κριτήρια Gini-Index, Cross-entropy καθώς είναι παραγωγίσιμα κάνοντάς τα καταλληλότερα για αριθμητικές βελτιστοποιήσεις. Επιπλέον είναι πιο ευαίσθητα σε αλλαγές των πιθανοτήτων σε σχέση με το Σφάλμα Λανθασμένης Ταξινόμησης. Για παράδειγμα, σε ένα πρόβλημα όπου η μεταβλητή απόκρισης έχει δύο επίπεδα($k = 2$) με 40 παρατηρήσεις σε κάθε κλάση, εάν υποθέσουμε ότι ένας διαχωρισμός δημιούργησε δύο κόμβους έναν (30,10) και (10,30) και έναν άλλο (20,40) και (20,0) και οι δύο διαχωρισμοί παρουσιάζουν Σφάλμα Λανθασμένης Ταξινόμησης 0.25, με τη μόνη διαφορά ότι ο δεύτερος διαχωρισμός δημιούργησε έναν κόμβο πιο καθαρό που είναι και πιθανότερα προτιμότερος. Το Gini-Index καθώς και το Cross-entropy είναι χαμηλότερα για τον δεύτερο διαχωρισμό. Για το λόγο αυτό συνήθως χρησμιποιούνται αυτά τα δύο κριτήρια για το μεγάλωμα ενός δέντρου. Για το κλάδεμα του δέντρου συνήθως χρησιμοποείται το κριτήριο της Λανθασμένης Ταξινόμησης. Στο Σχήμα 2.1 παρουσιάζονται τα τρία κριτήρια. Όπου με πράσινο χρώμα απεικονίζεται το κριτήριο Λανθασμένης Ταξινόμησης, με πορτοκαλί το Cross-entropy και με μπλε το Gini Index.




Σχήμα 2.1 Σύγκριση μεταξύ των τριών κριτηρίων διαχωρισμού

## 2.3 Η τεχνική Random Forest

Το πρόβλημα των μεθόδων οι οποίες βασίζονται σε δέντρα είναι ότι πολλές φορές παρουσιάζουν υπερπροσαρμογή στο σύνολο των δεδομένων εκπαίδευσης και δεν μπορούν να γενικευτούν σε άλλα σύνολα δεδομένων. Μία λύση του προβλήματος όπως αναφέραμε και στις δύο προηγούμενες ενότητες είναι το κλάδεμα του δέντρου. Το αρνητικό της συγκεκριμένης τεχνικής είναι ότι τα παραγώμενα δέντρα είτε θα υπερειδικεύονται πολύ στο σύνολο των δεδομένων(overfitting), είτε συρικνόντάς τα κατά πολύ δεν θα εκπαιδεύονται επαρκώς(underfitting) με αποτέλεσμα να οδηγούν σε μη ικανοποιητικά αποτελέσματα. Τη λύση στο πρόβλημα έδωσε ο Tin Kam Ho προτείνοντας την παραγωγή πολλών παραλλαγών του πρότοτυπου συνόλου δεδομένων με την μέθοδο Bootstrap, στα οποία εκπαίδευσε δέντρα όπου το τελικό μοντέλο συνδύαζε όλα τα δέντρα που είχαν δημιουργηθεί. Η συγκεκριμένη τεχνική ονομάζεται Bagging.

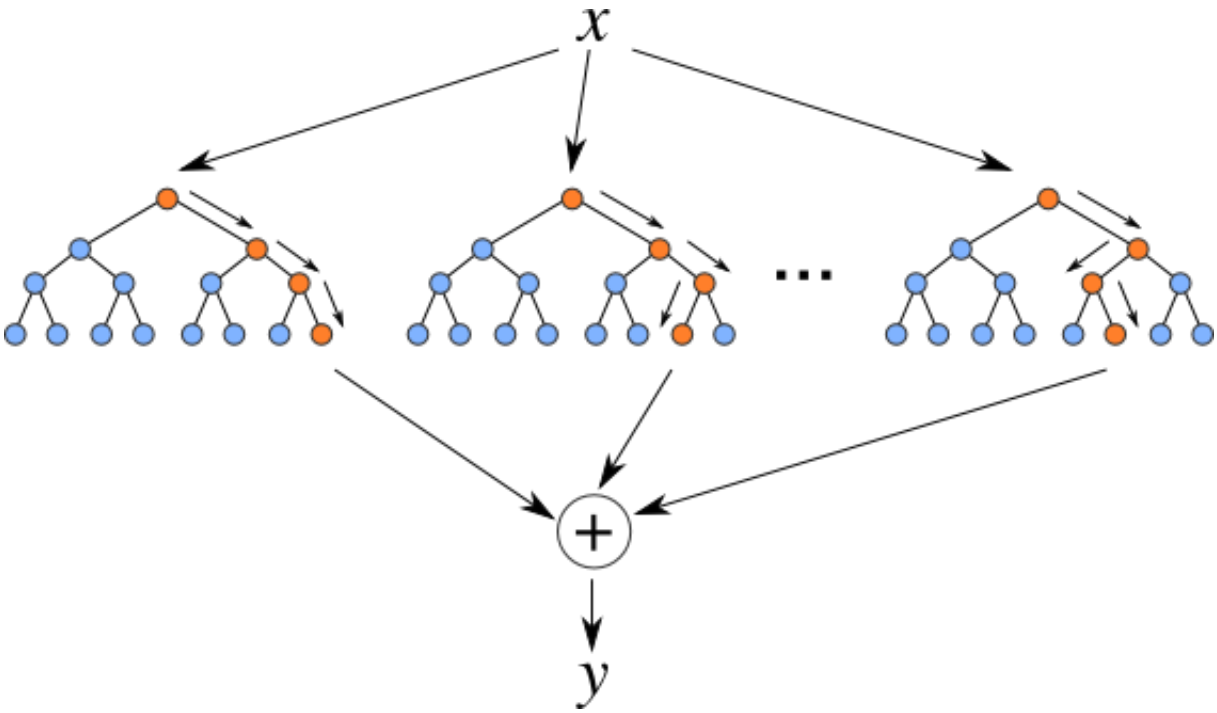
Μία επέκταση της τεχνικής αυτής προτάθηκε από τον Brieman(1997). Παρατήρησε ότι όσο πιό ασυσχέτιστες είναι οι προβλέψεις των δέντρων που δημιουργούνται τόσο πιό ικανοποιητικά αποτελέσματα αποφέρει το μοντέλο. Η συγκεκριμένη τεχνική λειτουργεί εώς εξής :

- Επιλέγεται ένα τυχαίο δείγμα μεγέθους $N$ με επανάθεση, δηλαδή μπορούμε να έχουμε μία παρατήρηση περισσότερο από μία φορές.
- Κατασκευάζεται ένα δέντρο που χρησιμοποιεί $m$ μεταβλητές ($m \subset p$), οι οποίες επιλέγονται τυχαία από τις συνολικά $p$ μεταβλητές.
- Η διαδικασία επαναλαμβάνεται Κ φορές, όπου Κ είναι μία παράμετρος που ορίζεται από τον χρήστη.
- Το τελικό μοντέλο συνδυάζει όλα τα παραγώμενα δέντρα παίρνοντας των μέσο όρο των προβλέψεων τους.

Επιλέγοντας τυχαία $m$ μεταβλητές με αυτήν την μέθοδο ο Brieman κατάφερε να κατασκευάζει δέντρα τα οποία είναι δοιαφορετικά μεταξύ τους. Η παράμετρος $m$ πρέπει να οριστεί εξ'αρχής από τον χρήστη. Όσο πιο μικρές τιμές παίρνει τόσο πιο ασυσχέτιστες είναι προβλέψεις των δέντρων μεταξύ τους. Χρησιμοποιώντας λίγες μεταβλητές υπάρχει κίνδυνος δημιουργίας μη

εκπαιδευμένων αρκετά δέντρων οπότε οδηγούμαστε σε μη αποδοτικό μοντέλο . Αυξάνοντας τον αριθμό $m$ των μεταβλητών που χρησιμοποιούμε για την δημιουργία των δέντρων αύξάνεται η απόδοση του κάθε δέντρου, αλλά πολλές φορές όχι η συνολική απόδοση του τελικού μοντέλου. Η συνήθης επιλογή της παραμέτρου είναι $m = \sqrt{p}$.

Στο Σχήμα 2.2 απεικονίζεται η τεχνική Random Forest.




Σχήμα 2.2 Εκπαίδευση δέντρων με την τεχνική Random Forest

## 2.4 Η τεχνική Boosting

Μία άλλη πολύ διαδεδομένη μέθοδος βασισμένη σε δέντρα είναι η Boosting. Η τεχνική λειτουργεί παρόμοια με την Random Forest, εκτός από το γεγονός ότι τα δέντρα δημιουργούνται διαδοχικά. Κάθε δέντρο εκπαιδεύεται χρησιμοποιώντας πληροφορία από τα προηγούμενα δέντρα, σε αντίθεση με την προηγούμενη μέθοδο που τα δέντρα είναι ασυσχέτιστα μεταξύ τους. Ο αλγόριθμος λειτουργεί εώς εξής:

- Θέτουμε $\hat{f}(x) = 0$ και τα κατάλοιπα $\varepsilon_i = y_i$ για κάθε παρατήρηση στο σύνολο των δεδομένων εκπαίδευσης.
- Εκπαιδεύουμε ένα δέντρο $\widehat{f^k}$ σε κάθε γύρο $k$ με $d$ κόμβους έχοντας σαν μεταβλητή απόκρισης τα κατάλοιπα.

- Προσθέτουμε μία περικομένη έκδοση του νέου δέντρου :

$$\hat{f}(x) \leftarrow \hat{f}(x) + \lambda \widehat{f^k}(x)$$

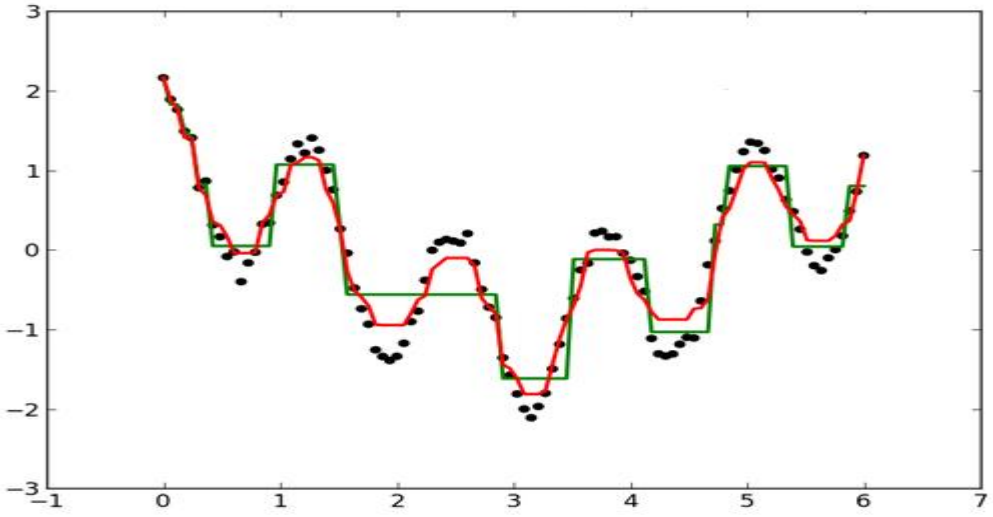
- Αναβαθμίζουμε τα κατάλοιπα :

$$\varepsilon_i \leftarrow \varepsilon_i - \lambda \widehat{f^k}(x)$$

- Επαναλαμβάνουμε την διαδικασία από το βήμα 2 $K$ φορές (καθορίζεται από τον χρήστη) καταλήγοντας στην τελική μορφή του μοντέλου:

$$\hat{f}(x) = \lambda \sum_{k=1}^{K} \widehat{f^k}(x)$$

Η εκπαίδευση ενός μεγάλου δέντρου είδαμε στις προηγούμενες ενότητες μπορεί να έχει ως αποτέλεσμα την υπερπροσαρμογή του μοντέλου. Η εκμάθηση Boosting γίνεται αργά και ελεγχόμενα. Έχουμε την επιλογή όταν δούμε ότι δεν προσφέρεται άλλη σημαντική πληροφορία να σταματήσουμε τη διαδικασία. Εκπαιδεύοντας μικρά δέντρα στα κατάλοιπα βελτιώνουμε την πρόβλεψη σε σημεία που δεν αποδίδει σωστά. Η παράμετρος $\lambda$ μειώνει τη διαδικασία εκμάθησης, επιτρέποντας σε διαφορετικά σχηματισμένα δέντρα να βλετιώσουν τις προβλέψεις και αντίστοιχα να μειώσουν τα κατάλοιπα. Στο Σχήμα 2.3 παρουσιάζεται η τεχνική Boosting, για διάφορες τιμές του $K$. Η πράσινη γραμμή είναι για $K$=1 και η κόκκινη για $K$=300.




Σχήμα 2.3 Εκπαίδευση Δέντρων με την μέθοδο Boosting

Η τεχνική Boosting γιά να είναι αποτελεσματική θα πρέπει ο χρήστης να ορίσει τον αριθμό των δέντρων που θα δημιουργηθούν, την παράμετρο $\lambda$ και τον αριθμό κόμβων σε κάθε δέντρο.Σε αντίθεση με την τεχνική Random Forest μεγάλος αριθμός δέντρων μπορεί εύκολα να υπερπροσαρμοστεί στα δεδομένα εκπαίδευσης και να μην μπορούν να αποδώσουν σε άλλα σύνολα δεδομένων. Η παράμετρος $\lambda$ ορίζει πόσο γρήγορα θα μαθαίνει το μοντέλο. Συνήθεις τιμές είναι από 0.001 εώς 0.1. Ο αριθμός των κόμβων ελέγχει την πολυπλοκότητα κάθε δέντρου. Συχνά δέντρα ενός διαχωρισμού γνωστά και ως "κλαδιά" είναι ικανοποιητικά καθώς το μοντέλο μας θέλουμε να μαθαίνει αργά.

## 2.5 Η τεχνική Gradient Boosting

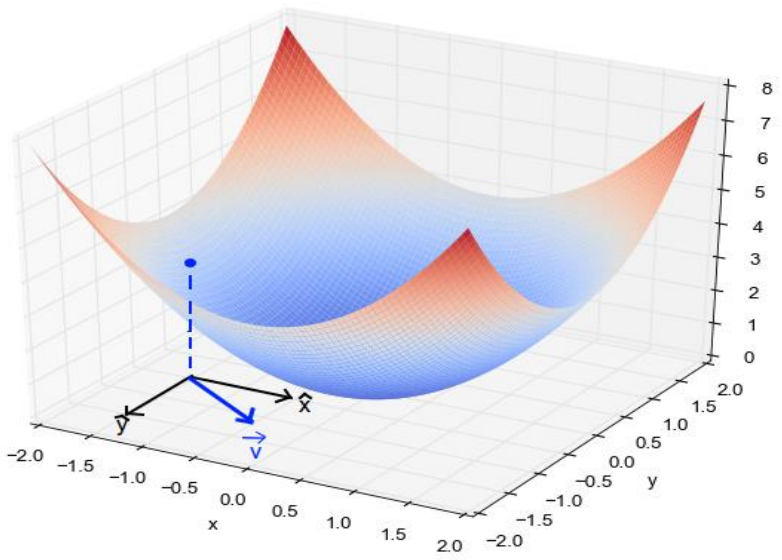
Η συγκεκριμένη τεχνική προτάθηκε από τον Friedman(2001) και ουσιαστικά αποτελεί μία επέκταση της τεχνικής Boosting. Το όνομα της το πήρε από δύο μεθόδους, τον αλγόριθμο Gradient Descent και την τεχνική Boosting.

Η Gradient Descent αποτελεί μία μέθοδο βελτιστοποίησης πρώτης τάξης. Για να βρει κάποιος το ολικό ελάχιστο μιας συνάρτησης χρησιμοποιώντας τη συγκεκριμένη τεχνική, αρχικά υπολογίζει την παράγωγό της και κάνει ανάποδα βήματα από την κατεύθυνση της παραγώγου. Η παράγωγος μετράει κατά πόσο θα αλλάξει η τιμή μίας συνάρτησης $J(\theta)$ εάν μεταβληθεί

ελάχιστα η μεταβλητή $\theta$. Ουσιαστικά είναι η κλίση της συνάρτησης, υψηλές τιμές της συνάρτησης υποδηλώνουν μεγάλη κλίση άρα και μεγάλη μεταβολή στην τιμή της $J(\theta)$ για μικρές μεταβολές του $\theta$.Ο συγκεκριμένος αλγόριθμος είναι επαναληπτικός αρχικοποιεί μία τυχαία τιμή στο $\theta$ υπολογίζει την παράγωγο της συνάρτησης στο συγκεκριμένο σημείο και μεταβάλει το $\theta$ κατά :

$$\theta = \theta - \rho \frac{dJ}{d\theta}$$

όπου η παράμετρος $\rho$ καθορίζει πόσο γρήγορα θα κινηθούμε στην αρνητική κατεύθυνση της παραγώγου. Η διαδικασία επαναλμβάνεται εώς ότου συγκλίνει ο αλγόριθμος. Ο αλγόριθμος παρουσιάζεται στο Σχήμα 2.4.




Σχήμα 2.4 Εύρεση ελάχιστου μέσω του αλγόριθμου Gradient Descent

Αυτό που πρότεινε ο Friedman είναι να εκπαιδεύουμε δέντρα στην αρνητική παράγωγο της συνάρτησης ζημιάς, σε αντίθεση με την μέθοδο Boosting όπου η εκπαίδεση γινόταν στα κατάλοιπα του προηγούμενου γύρου. Παίρνοντας για παράδειγμα ως συνάρτηση ζημιάς το άθροισμα των τετραγώνων των καταλοίπων διαιρεμένο με δύο :

$$L(y_i, \hat{y_i}) = \frac{1}{2}\sum_{i=1}^{N}(y_i - \hat{y_i})^2$$

και υπολογίζοντας την παράγωγο της παρατηρούμε ότι :

$$\frac{dL(y_i, \hat{y_i})}{d\hat{y_i}} = \hat{y_i} - y_i$$

Δηλαδή η αρνητική παράγωγος της συνάρτησης ζημιάς ισούται με τα κατάλοιπα. Οπότε εκπαιδεύουμε ένα δέντρο στα κατάλοιπα και προσθέτουμε μία περικομένη(κατά $\rho$) έκδοση του νέου δέντρου. Παρατηρούμε ότι χρησιμοποιώντας τη συγκεκριμένη συνάρτηση ζημιάς η τεχνική Gradient Boosting είναι ισοδύναμη με την Boosting.

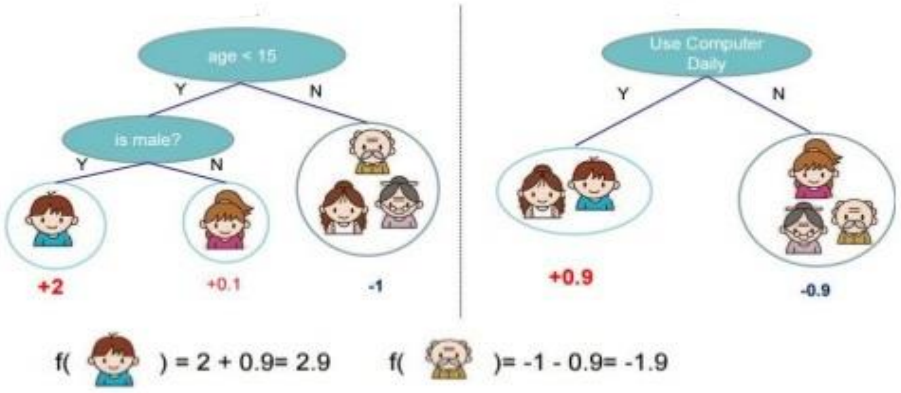
Ο λόγος που η μέθοδος Gradient Boosting είναι ισχυρότερη από αυτή του Boosting είναι ότι μας δίνει τη δυνατότητα επιλογής διαφορετικών συναρτήσεων ζημιάς. Ανάλογα με τη δομή των δεδομένων μας χρησιμοποιούνται διαφορετικές συναρτήσεις ζημιάς. Για παράδειγμα αν στα δεδομένα μας υπάρχουν ακραίες παρατηρήσεις επηρεάζεται πολύ περισσότερο το άθροισμα των τετραγώνων από ότι το άθροισμα των απόλυτων καταλοίπων από αυτές κάνοντας την χρήση του δεύτερου καταλληλότερη. Οι παράμετροι που πρέπει να οριστούν από τον χρήστη είναι ίδιες με αυτές της τεχνικής Boosting με τη μόνη διαφορά ότι δίνεται και η επιλογή συνάρτησης ζημιάς με προυπόθεση ότι η συνάρτηση που επιλέγεται είναι παραγωγίσιμη.

## 2.6 Η τεχνική XGBoost

Η τεχνική eXtreme Gradient Boosting προτάθηκε από τον Chen(2014). Όπως προαναφέραμε η τεχνική Gradient Boosting προσθέτει διαδοχικά δέντρα σε κάθε χρονική στιγμή $t$ στην αρνητική παράγωγο της συνάρτησης ζημιάς.

$$\hat{y_i}^{(t)} = \sum_{t=1}^{K} f_t(x_i)\,, f_t \in F \qquad (2.6.1)$$

όπου $F = \{f(x) = w_{q(x)}\}$ και $q: R^m \to T, w \in R^T$. Το $q$ αντιπροσωπεύει την δομή κάθε δέντρου. Το $T$ αντιπροσωπεύει τον αριθμό των φύλλων σε κάθε δέντρο. Κάθε $f_t$ αντιστοιχεί σε μία ανεξάρτητη δομή δέντρου $q$ και βάρη φύλλων $w$. Στο σχήμα 2.5 απεικονίζονται η τεχνική XGBoost η οποία συνδυάζει δύο δέντρα διαφορετικής δομής $q$. Η δομή ενός δέντρου ουσιαστικά είναι ο αριθμός των κόμβων που δημιουργούνται σε κάθε δέντρο.




Σχήμα 2.5 Ο αλγόριθμος XGBoost

Η συνάρτηση ζημιάς που ελαχιστοποιήσουμε σε κάθε χρονική στιγμή $t$ έχει τύπο :

$$L^{(t)} = \sum_{i=1}^{n} l\left(y_i, \hat{y_i}^{(t)}\right) + \sum_{k=1}^{T} \Omega_{f(t)}$$

Ο πρώτος όρος μετράει πόσο καλά προσαρμόζεται το μοντέλο στα δεδομένα εκπαίδευσης(μικρές τιμές υποδηλώνουν καλή προσαρμοργη) και ο δεύτερος την πολυπλοκότητα του κάθε δέντρου. Στην πολυπλοκότητα του δέντρου εισάγεται και ένας νέος όρος εκτός από τον αριθμό φύλλων$(T)$, πιο συγκεκριμένα γίνεται συρρίκνωση των βαρών των φύλλων.

$$\Omega_{f(t)} = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2$$

Η παράμετρος $\gamma$ υποδηλώνει πόσο θα τιμωρούμε το μεγάλωμα του δέντρου. Μεγάλες τιμές του $\gamma$ θα μας οδηγήσουν σε μικρά δέντρα, αντίστοιχα μικρές τιμές του $\gamma$ θα μας οδηγήσουν σε μεγάλα δέντρα. Η παράμετρος $\lambda$ ρυθμίζει κατά πόσο θα συρρικνώνονται τα βάρη του δέντρου. Όσο αυξάνεται η τιμή της τα βάρη του δέντρου συρρικώνονται. Από την σχέση (2.6.1) έχουμε ότι :

$$\widehat{y_i}^{(t)} = \sum_{t=1}^{K} f_t(x_i) = \widehat{y_i}^{(t\text{-}1)} + f_t(x_i)$$

Οπότε πρέπει να αποφασίσουμε ποιό $f_t(x_i)$ ελαχιστοποιεί τη συνάρτηση ζημιάς τη χρονική στιγμή t :

$$L^{(t)} = \sum_{i=1}^{n} l\left(y_i, \widehat{y_i}^{(t)}\right) + \sum_{k=1}^{T} \Omega_{f(t)}$$

$$= \sum_{i=1}^{n} l\left(y_i, \widehat{y_i}^{(t\text{-}1)} + f_t(x_i)\right) + \sum_{k=1}^{T} \Omega_{f(t)}$$

Από το ανάπτυγμα Taylor έχουμε :

$$f(x + \Delta x) \cong f(x) + f'(x)\Delta x + \frac{1}{2} f''(x)(\Delta \chi)^2$$

Άρα,

$$L^{(t)} \cong \sum_{i=1}^{n} \left[ l\left(y_i, \widehat{y_i}^{(t\text{-}1)}\right) + g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i) \right] + \Omega_{f(t)}$$

Όπου $g_i = d_{\widehat{y_i}^{(t\text{-}1)}} l\left(y_i, \widehat{y_i}^{(t\text{-}1)}\right)$ και $h_i = d^2_{\widehat{y_i}^{(t\text{-}1)}} l\left(y_i, \widehat{y_i}^{(t\text{-}1)}\right)$

Αφαιρώντας τις σταθερές η συνάρτηση ζημιάς γίνεται :

$$L'^{(t)} = \sum_{i=1}^{n} \left[ g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i) \right] + \Omega_{f(t)}$$

Ορίζοντας $I_j = \{ i | g(x\_i) = j \}$ το σετ των παρατηρήσεων στο φύλλο $j$ μπορούμε να ξανά γράψουμε την παραπάνω σχέση ως εξής :

$$L'^{(t)} = \sum_{i=1}^{n} \left[ g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i) \right] + \Omega_{f(t)}$$

$$= \sum_{i=1}^{n} \left[ g_i w_{q(x_i)} + \frac{1}{2} h_i w_{q(x_i)}^2 \right] + \Omega_{f(t)}$$

$$= \sum_{i=1}^{T} \left[ \left( \sum_{i \in I_j} g_i \right) w_j + \frac{1}{2} \left( \sum_{i \in I_j} h_i + \lambda \right) w_j^2 \right] + \gamma T$$

Θέτοντας $G_j = \sum_{i \in I_j} g_i$ και $H_j = \sum_{i \in I_j} h_i$ προκύπτει :

$$L'^{(t)} = \sum_{j=1}^{T} \left[ G_j w_j + \frac{1}{2} \left( H_j + \lambda \right) w_j^2 \right] + \gamma T \qquad (2.6.2)$$

Υποθέτοντας ότι η δομή του δέντρου $(q(x))$ είναι δεδομένη, το βέλτιστο βάρος σε κάθε φύλλο βρίσκεται ελαχιστοποιώντας ως προς $w_j$ την σχέση (2.6.2).

$$w_j^* = -\frac{G_j}{H_j + \lambda}$$

Αντικαθιστώντας το $w_j^*$ στην σχέση (2.6.2) έχουμε $L'^{(t)} = -\frac{1}{2} \sum_{j=1}^{T} \frac{G_j^2}{H_j + \lambda} + \gamma T$ το οποίο μετράει πόσο καλή είναι η δομή του νέου δέντρου. Τέλος ο αλγόριθμος δημιουργεί διαχωρισμούς χρησιμοποιώντας την εξής συνάρτηση:

$$Gain = \frac{1}{2} \left[ \frac{G_L^2}{H_L + \lambda} + \frac{G_R^2}{H_R + \lambda} - \frac{(G_L + G_R)^2}{H_L + H_R + \lambda} \right] - \gamma$$

όπου το πρώτο κλάσμα είναι το σκορ αριστερού μέρους του διαχωρισμού, το δεύτερο είναι το σκορ του δεξιού μέρους του διαχωρισμού, το τρίτο είναι το σκορ εάν δεν πραγματοποιηθεί ο διαχωρισμός και το $\gamma$ μετράει το κόστος πολυπλοκότητας του διαχωρισμού.

Όπως και στις άλλες τεχνικές ξεκινάμε με ένα δέντρο και το μεγαλώνουμε μέχρι ένα συγκεκριμένο βάθος που ορίζεται από τον χρήστη. Πραγματοποιείται κλάδεμα του δέντρου σε όσους διαχωρισμούς έχουν αρνητικό Gain. Τέλος προστήθεται στο μοντέλο περικομένη έκδοση κατά $\varepsilon$ του νέου δέντρου και επαναλαμβάνουμε τη διαδικασία Κ φορές.

Και σε αυτήν την μέθοδο υπάρχουν παράμετροι που πρέπει να οριστούν από τον χρήστη πριν το ξεκίνημα της διαδικασίας. Η παράμετρος $\varepsilon$ ρυθμίζει το συρρίκνωμα που θα έχει το κάθε δέντρο. Το μέγιστο βάθος είναι μία άλλη παράμετρος που ρυθμίζει των αριθμό των κόμβων σε ένα δέντρο. Η παράμετρος $\gamma$ ρυθμίζει την πολυπλοκότητα των δέντρων. Η παράμετρος $\lambda$ ρυθμίζει τη συρρίκνωση των βαρών των φύλλων του δέντρου. Υπάρχει δυνατότητα επιλογής συνάρτησης ζημιάς όπως και στην Gradient Boosting τεχνική. Τέλος πρέπει να οριστεί ο αριθμός $K$ των επαναλήψεων της διαδικασίας. Η συγκεκριμένη τεχνική έχει πάρα πολλές παραμέτρους αλλά ρυθμίζοντας τες σωστά είναι πολύ ανώτερη από όσες έχουμε αναφέρει.

## 2.7 Η τεχνική LightGBM

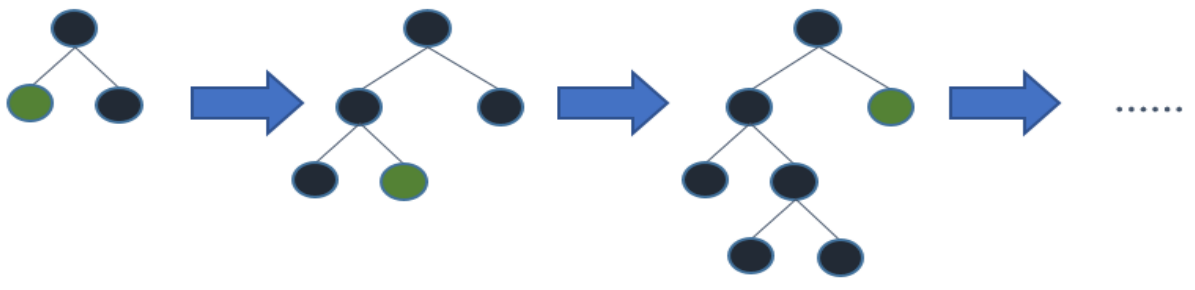
Η συγκεκριμένη τεχνική είναι παρόμοια με την XGBoost, έχοντας ελάχιστες διαφορές. Η πρώτη είναι στην διαδικασία που ακολουθεί ώστε να βρει τον καλύτερο διαχωρισμό. Η τεχνική XGBoost καθώς και όλες όσες αναφέραμε, σαρώνουν το σύνολο των δεδομένων αναζητώντας τον βέλτιστο διαχωρισμό. Λαμβάνοντας υπόψη τον όγκο δεδομένων που έχουμε διαθέσιμο σήμερα μία τέτοια προσέγγιση πιθανότατα είναι υπολογιστικά ακριβή. Ο συγκεκριμένος αλγόριθμος δημιουργεί ιστογράμματα και χρησιμοποιεί τις κλάσεις που παράγονται, αντί για όλο το εύρος τιμών της κάθε μεταβλητής, επιτυνχάνοντας σημαντική μείωση στο χρόνο εκπαίδευσης.

Η δεύτερη διαφορά είναι ότι η LightGBM δεν χρησιμοποιεί το σύνολο των δεδομένων εκπαίδευσης, αλλά ένα δείγμα του, το οποίο προκύπτει βάσει της Gradient-One-

Sampling(GOSS) μεθόδου. Η ιδέα είναι πως όλες οι παρατηρήσεις δεν συνισφέρουν το ίδιο στην εκπαίδευση του αλγορίθμου, καθώς όσες έχουν μικρή πρώτη παράγωγο συνάρτησης ζημιάς είναι πιο καλά εκπαιδευμένες από όσες έχουν μεγάλη. Μία προσέγγιση θα ήταν να αγνοήσουμε αυτές με μικρή παράγωγο, έχοντας ως αποτέλεσμα την δημιουργία μεροληπτικών δειγμάτων και την αλλαγή στην κατανομή των δεδομένων μας. Για παράδειγμα σε ένα σύνολο δεδομένων παρατηρήσεις που είναι ηλικιακά μικρότερες έχουν την τάση να εκπαιδεύονται καλύτερα,αγνοώντας τις μικρότερες ηλικίες το δείγμα που θα πάρουμε θα έχει μεγαλύτερη κατανομή ηλικίας. Θα οδηγηθούμε σε διαχωρισμό μεγαλύτερο από ότι ο βέλτιστος καθώς έχουμε παρατηρήσεις ηλικιακά μεγάλες μόνο, οδηγώντας στην υπερπροσαρμογή του μοντέλου μας στο δείγμα. Για να αντιμετωπιστεί το πρόβλημα επιλέγουμε επίσης τυχαία παρατηρήσεις με μικρή παράγωγο συνάρτησης ζημιάς. Συγκεκριμένα, ταξινομούνται τα δεδομένα σύμφωνα με την απόλυτη τιμή της παραγώγου τους, επιλέγονται οι $\alpha * 100\%$ με την μεγαλύτερη παράγωγο και $b * 100\%$ από τις υπόλοιπες.

Για τον υπολογισμό της συνάρτησης Gain οι παρατηρήσεις με μικρή παράγωγο πολλαπλασιάζονται με $\frac{1-a}{b}$ δίνοντας μεγαλύτερη σημασία στις μη καλά εκπαιδευμένες χωρίς να διαφοροποιείται η κατανομή των δεδομένων μας κατά πολύ. Εκπαιδεύοντας μόνο ένα δείγμα σε κάθε επανάληψη επιτυγχάνεται σημαντική αύξηση στη διαδικασία εκμάθησης του αλγορίθμου, έχοντας ως αποτέλεσμα τη σύγκλισή του πιο γρήγορα στη βέλτιστη λύση.

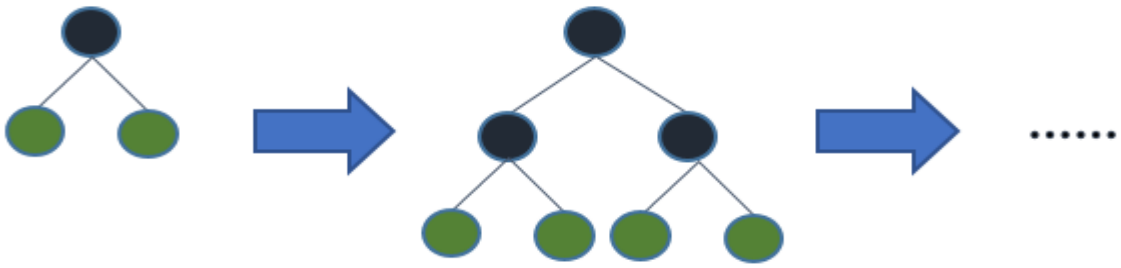
Μία άλλη διαφορά είναι στο πως η συγκεκριμένη μέθοδος επιλέγει να μεγαλώσει ένα δέντρο. Όσες τεχνικές αναφέραμε προς το παρών μεγαλώνουν δέντρα ψάχνοντας τον βέλτιστο διαχωρισμό, κρατώντας τον αριθμό κόμβων σε κάθε επίπεδο του δέντρου ίδιο. Στο Σχήμα 2.6 φαίνεται το μεγάλωμα ενός δέντρο βάσει της συγκεκριμένης τεχνικής.



Σχήμα 2.6 Μεγάλωμα δέντρου ανά επίπεδα

Αρχικά δημιουργούνται δύο κόμβοι και στη συνέχεια δημιουργούνται όλοι οι δυνατοί κόμβοι στο πρώτο επίπεδο του δέντρου. Στη συνέχεια πραγματοποιούνται όλοι οι δυνατοί συνδυασμοί στο δεύτερο επίπεδο, η διαδικασία συνεχίζεται μεχρί ώτου το δέντρος φτάσει σε βάθος ίσο με αυτό που έχει ορίσει ο χρήστης και στη συνέχεια το δέντρο κλαδεύεται.

Η τεχνική LightGBM μεγαλώνει δέντρα ψάχνοντας τον βέλτιστο διαχωρισμό σε κάθε τερματικό κόμβο ανεξαρτήτως επιπέδου. Ουσιαστικά εξετάζει κάθε φορά ποιός τερματικός κόμβος αποφέρει τη μεγαλύτερη μείωση στη συνάρτηση ζημιάς. Στο Σχήμα 2.7 παρουσιάζεται αναλυτικά το μεγάλωμα ενός δέντρου με την συγκεκριμένη τεχνική.



Σχήμα 2.7 Μεγάλωμα δέντρου ανά τερματικό κόμβο

Πρέπει να συμειωθεί ότι η συγκεκριμένη τεχνική είναι πολύ ευαίσθητη και μπορεί να οδηγήσει σε υπερπροσαρμογή του αλγορίθμου στο σύνολο δεδομένων που εκπαιδεύτηκε, κάνοντας τον μη αποτελεσματικό σε νέα σύνολα δεδομένων. Η επιλογή του μέγισοτυ βάθους που αρχικοποείται από τον χρήστη είναι πολύ σημαντική, καθώς σωστή επιλογή συνήθως οδηγεί σε πολύ καλύτερα αποτελέσματα από ότι το μεγάλωμα δέντρου ανά επίπεδα. Λανθασμένη επιλογή οδηγεί πολύ εύκολα σε υπερπροσαρμογή (overfit). Τέλος η LightGBM χρησιμοποιεί την τεχνική Exclusive Feature Bundling (EFB). Η συγκεκριμένη τεχνική προσπαθεί να μειώσει τις διαστάσεις των κατηγορικών μεταβλητών "δένοντας" (bundle) όσων τα επίπεδα των παραγότνων είναι μηδενικά ταυτόχρονα. Για παράδειγμα σε ένα συνολο δεδομένων μπορεί να μην υπάρχει παρατήρηση που να είναι "Άντρας" και "Ψηλός", όποτε ενοποιώντας τες σε μία να μην χάνεται καμία πληροφορία. Εκτός από τη μείωση των διαστάσεων των μεταβλητών με την συγκεκριμένη προσέγγιση επιταχύνεται η διαδικασία εκμάθησης του αλγορίθμου και πολλές φορές γενικεύεται καλύτερα. Οι διαφορές που εντοπίζονται στις δύο τεχνικές

(LightGBM,XGBoost) είναι ελάχιστες, στην πράξη αποφέρουν σχεδόν παρόμοια αποτελέσματα. Η μόνη σημαντική διαφοροποίηση είναι στο υπολογιστικό κόστος με την LightGBM να είναι σαφώς "οικονομικότερη", εξού και η ονομασία της Light.

# 3. Γραμμικά Μοντέλα

## 3.1 Λογιστική Παλινδρόμηση

Η λογστική παλινδρόμηση χρησιμοποιείται για την κατηγοριοποίηση κατηγορικών μεταβλητών δοθέντος ενός συνόλου ανεξάρτητων μεταβλητών. Πολλές φορές συγκρίνεται με την διαχωριστική ανάλυση με την πρώτη συνήθως να αποφέρει καλύτερα αποτελέσματα καθώς δεν υποθέτει ότι οι επεξηγηματικές μεταβλητές ακολουθούν κανονική κατανομή.

Στο μοντέλο της λογιστικής παλινδρόμησης η μεταβλητή απόκρισης παλινδρομείται σε ένα σύνολο ανεξάρτητων μεταβλητών $m$ $X_1, X_2, \dots, X_m$. Το σύνολο τιμών της μεταβλητής απόκρισης είναι το $\{0,1\}$. Παλινδρομώντας τις ανεξάρτητες μεταβλητές οι εκτιμήσεις που παίρνουμε έχουν πεδίο τιμών το $(-\infty, +\infty)$. Για το λόγο αυτό χρησιμοποιείται η συνάρτηση $\text{logit}(\pi)$ η οποία έχει πεδίο ορισμού το $[0,1]$ και πεδίο τιμών όλο το $R$. Για $\pi \in [0,1]$ έχουμε ότι :

$$logit(\hat{\pi}) = log\left(\frac{\hat{\pi}}{1-\hat{\pi}}\right)$$

Το μοντέλο της λογιστικής παλιδρόμησης δίνεται από τον παρακάτων τύπο:

$$logit(\hat{\pi}) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_m X_m$$

και η εκτιμώμενη πιθανότητα π υπολογίζεται εώς :

$$\hat{\pi} = \frac{exp(\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_m X_m)}{1 + exp(\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_m X_m)}$$

Η μόνη υπόθεση που γίνεται είναι πως όλες οι παρατηρήσεις είναι ανεξάρτητες μεταξύ τους και ακολουθούν κατανομή $Bernoulli(\pi)$. Οι εκτιμήσεις των παραμέτρων $\beta$ γίνονται από την μεγιστοποίηση της πιθανοφάνειας της κατανομής Bernoulli.

$$L(\beta/X) = \prod_{i=1}^{N} Pr(Y/X;\beta)$$

$$\prod_{i=1}^{N} \hat{\pi}^{y_i} (1-\hat{\pi})^{1-y_i}$$

υπολογίζοντας τον λογάριθμο της πιθανοφάνειας έχουμε ότι :

$$logL(\beta/X) = \sum_{i=1}^{N} \hat{\pi}^{y_i} (1-\hat{\pi})^{1-y_i}$$

Δεν υπάρχει κλειστός τύπος για τον υπολογισμό των παραμέτρων που μεγιστοποιούν την παραπάνω σχέση. Η μέθοδος με την οποία μεγιστοποιείται η παραπάνω σχέση είναι αυτή των Newton-Raphson. Είναι παρόμοια με την Gradient Descent που περιγράψαμε σε προηγούμενη ενότητα.

- Αρχικοποιείται μία τυχαία τιμή $x_0$.
- Υπολογίζονται τα $f(x_0)$ και $f'(x_0)$.
- Στη συνέχεια υπολογίζεται το επόμενο σημείο ως :

$$x_2 = x_0 + \frac{f(x_0)}{f'(x_0)}$$

- Η διαδικασία επαναλαμβάνεται από το δεύτερο βήμα εως ότου συγκλίνει ο αλγόριθμος σε κάποιο μέγιστο ολικό ή τοπικό, είτε ικανοποιηθεί κάποιο κριτήριο επαναλήψεων που ορίζεται από τον χρήστη.

## 3.2 Ridge Παλινδρόμηση

Η τεχνική Ridge Παλινδρόμηση δημιουργήθηκε με σκόπο την αντιμετώπιση του φαινομένου της πολυσυγγραμικότητας. Το φαινόμενο της πολυσυγγραμικότητας

εμφανίζεται σε σύνολα δεδομένων όπου οι επεξηγηματικές μεταβλητές παρουσιάζουν υψηλή γραμμική συσχέτιση. Στο μοντέλο της γραμμικής παλινδρόμησης έχοντας επεξηγηματικές μεταβλητές υψηλά συσχετισμένες, ανεξαρτήτως εάν οι ισχύουν οι προϋποθέσεις του γραμμικού μοντέλου δηλαδή οι εκτίμήτριες ελαχίστων τετραγώνων είναι οι καλύτερες γραμμικές αμερόληπτες εκτιμήτριες, οδηγούμαστε σε υψηλά τυπικά σφάλματα για τις εκτιμήτριες ελαχίστων τετραγώνων.

Οι εκτιμητές ελαχίστων τετραγώνων των $\beta$ είναι :

$$\hat{\beta}^{ols} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y}$$

Για τον υπολογισμό των εκτιμητριών ελαχίστων τετραγώνων χρειάζεται ο υπολογισμός του $(\boldsymbol{X}'\boldsymbol{X})^{-1}$. Χρησιμοποιώντας τον τύπο :

$$A^{-1} = \frac{1}{det(A)} adj(A)$$

παρατηρούμε ότι ο αντίστροφος ενός πίνακα είναι ο συμπληρωματικός πίνακας διαιρεμένος με την ορίζουσα του πίνακα. Στην περίπτωση της πολυσυγραμμικότητας η ορίζουσα του πίνακα $\boldsymbol{X}'\boldsymbol{X}$ είναι πολύ κοντά στο μηδέν έχοντας ως αποτέλεσμα την αύξηση των παραμέτρων του γραμμικού μοντέλου.

Αυτό που πρότειναν οι Hoerl και Kennard ήταν τη συρρίκνωση των παραμέτρων προσθέτοντας έναν όρο στη συνάρτση ζημιάς ώστε να τιμωρούνται μεγάλες τιμές των παραμέτρων. Η συνάρτηση ζημιάς χρησιμοποιώντας ως επεξηγηματικές τις κεντροκοποιημένες μεταβλητές έχει τον εξής τύπο :

$$\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 + \lambda \sum_{k=1}^{p} \beta_k^2$$

υπό τον περιορισμό

$$\sum_{k=1}^{p} \beta_k^2 \leq t$$

Οι παράμετροι της Ridge παλινδρόμησης εκτιμούνται εώς :

$$\hat{\beta}^{ridge} = (\mathbf{X}'\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}'\mathbf{y}$$

όπου ο πίνακας $\boldsymbol{I}$ είναι ο μοναδιαίος $p \times p$. Ουσιαστικά προστήθεται μία θετική σταθερά στα διαγώνια στοιχεία του πίνακα $\boldsymbol{X'X}$ πριν την αντιστροφή του. Στην περίπτωση όπου τα διανύσματα των μεταβλητών $X_k$ είναι ορθογώνια έχουμε ότι $\hat{\beta}^{ols} = \boldsymbol{X'y}$ , καθώς $\boldsymbol{X'X} = \boldsymbol{I}$. Οπότε οι συντελεστές της παλινδρόμησης Ridge είναι ίσοι με :

$$\hat{\beta}^{ridge} = (\mathbf{X'X} + \lambda\mathbf{I})^{-1}\mathbf{X'y}$$

$$\hat{\beta}^{ridge} = (\mathbf{I} + \lambda\mathbf{I})^{-1}\hat{\beta}^{ols}$$

Από την παραπάνω σχέση προκύπτει :

$$\hat{\beta}^{ridge} = \frac{1}{1+\lambda}\hat{\beta}^{ols}$$

οπού $\hat{\beta}^{ols}$ είναι οι εκτιμητές β που προκύπτουν από τα ελάχιστα τετράγωνα. Εύκολα αποδεικνύεται ότι οι εκτιμητές Ridge δεν είναι αμερόληπτοι εκτιμητές καθώς :

$$\mathrm{E}\left[\hat{\beta}^{ridge}\right] = \frac{1}{1+\lambda}\mathrm{E}\left[\hat{\beta}^{ols}\right] = \frac{1}{1+\lambda}\beta$$

Η διακύμναση όμως των εκτιμητών Ridge είναι μικρότερη από ότι των ελαχίστων τετραγώνων.

$$V\left[\hat{\beta}^{ridge}\right] = \left(\frac{1}{1+\lambda}\right)^2 V\left[\hat{\beta}^{ols}\right]$$

Ουσιαστικά η παράμετρος $\lambda$ είναι μια παράμετρος που ρυθμίζει το αντιστάθμισα μεταξύ μεροληψίας και διακύμανσης των εκτιμήτριων μας. Μεγάλες τιμές του $\lambda$ οδηγούν σε εκτιμητές με μικρή διακύμανση αλλά μεγάλη διακύμναση. Για μικρές τιμές του $\lambda$ οδηγούμαστε σε εκτιμήτριες με μεγάλη διακύμανση και μικρή μεροληψία. Τέλος το $\lambda = 0$ οι εκτιμήτριες που προκύπτουν είναι αυτές των ελαχίστων τετραγώνων. Σύνηθεις τιμές του $\lambda$ είναι μεταξύ 0 και 1.

## 3.3 Lasso Παλινδρόμηση

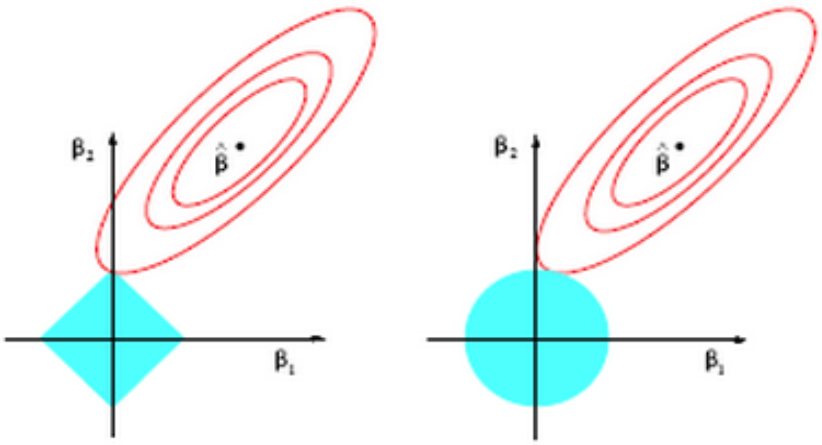
Μία επέκταση της τεχνικής Ridge είναι αυτή της Lasso παλινδρόμησης. Οι δύο τεχνικές είναι παρόμοιες με τη μόνη διαφορά ότι η παλινδρόμηση εκτιμάει τα β ελαχιστοποιώντας την εξής συνάρτηση ζημιάς :

$$\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 + \lambda \sum_{k=1}^{p}|\beta_k|$$

υπό τον περιορισμό

$$\sum_{k=1}^{p}|\beta_k| \leq t$$

Το πλεονέκτημα της συγκεκριμένης μεθόδου είναι ότι εκτός από την συρρίκνωση των βαρών ορισμένοι συντελεστές γίνονται ίσοι με το μηδέν καθώς η ρυθμιστική παράμετρος $\lambda$ αυξάνεται. Στην Ridge παλινδρόμηση για μεγάλες τιμές του $\lambda$ οι συντελεστές $\beta$ πηγαίνουν ασυμπτωτικά στο μηδέν χωρίς να μηδενίζονται. Με τον τρόπο αυτό γίνεται και επιλογή των πιο σημαντικών μεταβλητών για την πρόβλεψη της μεταβλητής απόκρισης. Στο Σχήμα 3.1 απεικονίζονται οι δύο μεθόδοι για την εκτίμηση των συντελεστών $\beta$, όπου $p = 2$.




Σχήμα 3.1 Μέθοδοι Ridge και Lasso παλινδρόμησης

Αριστερά απεικονίζεται η μέθοδος Lasso και δεξιά η Ridge παλινδρόμηση. Οι ελλείψεις του σχήματος είναι τα σημεία που η συνάρτηση ζημιάς έχει την ίδια τιμή η οποία γίνεται μικρότερη όσο πλησιάζουμε το κέντρο. Οι διαφορές που έχουν οι δύο μεθόδοι είναι στην περιοχή που ορίζεται από τον περιορισμό για τις τιμές των παραμέτρων. Ο περιορισμός της Lasso είναι $\sum_{k=1}^{p}|\beta_k| \leq t$,ο οποίος παριστάνει ένα ρόμβο στις δύο διαστάσεις. Από το Σχήμα 3.1 παρατηρούμε ότι ο συντελεστής $\beta_1$ γίνεται μηδέν στην βέλτιστη λύση της συνάρτησης ζημιάς.

Αντίθετα η μέθοδος Ridge που έχει περιορισμό $\sum_{k=1}^{p} \beta_k^2 \leq t$, ο οποίος παριστάνει κύκλους εκτιμά τον συντελεστή $\beta_1$ πολύ κοντά στο μηδέν.

## 3.4 Τεχνικές Ridge και Lasso στην Λογιστική Παλινδρόμηση

Οι δύο αυτές τεχνικές έχουν εφαρμογές και στην λογιστική παλινδρόμηση. Η Ridge παλινδρόμηση έχει τύπο εως εξής :

$$logit(\hat{\pi}) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_m X_m + \lambda \sum_{k=1}^{p} \beta_k^2$$

Η πιθανότητα $\hat{\pi}$ υπολογίζεται ως εξής :

$$\hat{\pi} = \frac{exp\left(\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_m X_m + \lambda \sum_{k=1}^{p} \beta_k^2\right)}{1 + exp\left(\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_m X_m + \lambda \sum_{k=1}^{p} \beta_k^2\right)}$$

Αντίστοιχα η τεχνική Lasso στην λογιστική παλινδρόμηση έχει τον εξής τύπο :

$$logit(\hat{\pi}) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_m X_m + \lambda \sum_{k=1}^{p} |\beta_k|$$

Η πιθανότητα $\hat{\pi}$ υπολογίζεται ως εξής :

$$\hat{\pi} = \frac{exp\left(\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_m X_m + \lambda \sum_{k=1}^{p} |\beta_k|\right)}{1 + exp\left(\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_m X_m + \lambda \sum_{k=1}^{p} |\beta_k|\right)}$$

Η εκτίμηση των παραμέτρων όπως και στην απλή λογιστική παλινδρόμηση γίνεται μεγιστοποιώντας την πιθανοφάνεια της κατανομής Bernoulli χρησιμοποιώντας τα αντίστοιχα $\hat{\pi}$. Όπως και στην γραμμική παλινδρόμηση έτσι και στην λογιστική οι τεχνικές Ridge και Lasso συντελούν στην μείωση της διακύμανσης των εκτιμητών β. Επίσης πρέπει να αρχικοποιηθεί τιμή στην παράμετρο $\lambda$. Σύνηθεις τιμές είναι μεταξύ 0 και 1.

# 4. Μέθοδοι Μείωσης Διαστάσεων

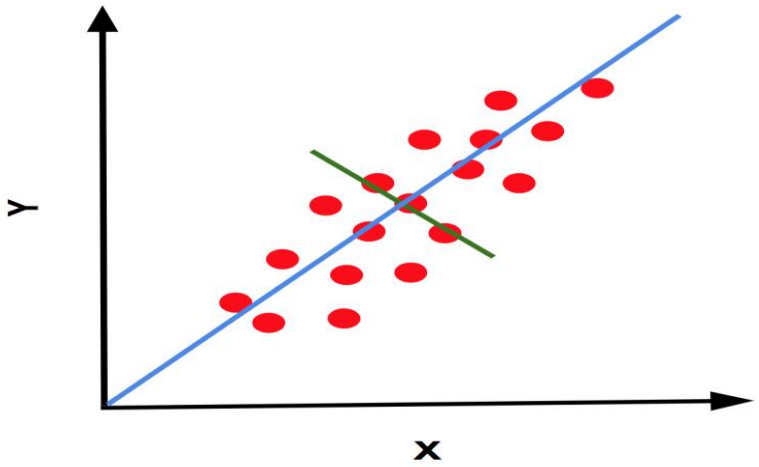
## 4.1 Ανάλυση Κύριων Συνιστωσών

Η Ανάλυση Κύριων Συνιστωσών προτάθηκε το 1901 από τον Pearson. Η συγκεκριμένη μέθοδος έχει ως στόχο την δημιουργία ασυσχέτιστων γραμμικών συνδυασμών των αρχικών μεταβλητών προσπαθώντας να κρατήσουν όσο το δυνατόν περισσότερη πληροφορία. Τα οφέλη της συγκεκριμένης τεχνικής δεν είναι μονο υπολογιστικά. Σε πολλά σύνολα δεδομένων οι μεταβλητές συνήθως είναι συσχετισμένες μεταξύ τους, πολλές φορές οδηγώντας στην υπερπροσαρμογή του μοντέλου. Έχοντας μεταβλητές ασυσχέτιστες μεταξύ τους το μοντέλο μας γενικεύεται πολύ καλύτερα σε νέα σύνολα δεδομένων.

Για την υλοποίηση της συγκεκριμένης μεθόδου χρείαζεται να υπολογιστεί ο πίνακας συνδιακυμάνσεων των μεταβλητών. Έστω Σ ο πίνακας συνδιακυμάνσεων των $p$ μεταβλητών. Ο πίνακας Σ είναι θετικά ημιορισμένος και συμμετρικός. Σύμφωνα με το φασματικό θεώρημα υπάρχουν $e_1, e_2, \dots, e_p$ ιδιοδιανύσματα και $\lambda_1, \lambda_2, \dots, \lambda_p$ ιδοτιμές τέτοια ώστε :

$$\Sigma = \lambda_1 e_1 e_1' + \lambda_2 e_2 e_2' + \cdots + \lambda_p e_p e_p'$$

Κάθε ένα από τα ζευγάρια $e_i$ αποτελεί μία ορθογώνια προβολή, το οποίο σημαίνει ότι για κάθε ζεύγος $e_i, e_j$ με $i \neq j$ είναι μεταξύ τους κάθετα οπότε και ασυσχέτιστα. Η εφαρμογή της μεθόδους φαίνεται στο Σχήμα 4.1 όπου η μπλέ γραμμή απικονίζει την πρώτη κύρια συνιστώσα και η πράσινη τη δεύτερη.




Σχήμα 4.1 Η Μέθοδος PCA

Η συνολική διασπορά του πίνακα των συνδιακυμάνσεων Σ αποτελείται από το άθροισμα των ιδιοτιμών $\lambda_i, i = 1,2,..,p$. Έστω $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_p \geq 0$ έχουμε ότι :

Το ποσοστό που ερμηνεύεται από την πρώτη κύρια συνιστώσα ισούται με :$\frac{\lambda_1}{\lambda_1+\lambda_2+\cdots+\lambda_p}$

Το ποσοστο που ερμηνευεται από τις δύο πρώτες μαζί ισούται με : $\frac{\lambda_1+\lambda_2}{\lambda_1+\lambda_2+\cdots+\lambda_p}$

Και προφανώς το ποσοστό που ερμηνεύουν όλες μαζί ισούται με : $\frac{\lambda_1+\lambda_2+\cdots+\lambda_p}{\lambda_1+\lambda_2+\cdots+\lambda_p} = 1$

Το ερώτημα που γεννάται είναι ποιός είναι ο βέλτιστος αριθμός συνιστωσών που πρέπει να χρησιμοποιηθεί. Ένα κριτήριο είναι να ορίσουμε ένα όριο για παράδειγμα 85% και να επιλέξουμε τον αριθμό των συνιστωσών που όλες μαζί εξηγούν μεγαλύτερο ποσοστό από το κατώφλι που ορίσαμε. Ένα αλλο κριτήριο είναι αυτό του Kaiser ο οποίος πρότεινε να κρατήσουμε μόνο όσες ιδιοτιμές είναι μεγαλύτερες από την μέση τιμή των ιδιοτιμών $\lambda_1, \lambda_2, \dots, \lambda_p$. Τέλος μπορούμε να αποφανθούμε για τον αριθμό των συνιστωσών σκεφτόμενοι το πρόβλημα που αναζητάμε να λύσουμε. Θα ξεκινήσουμε με ένα τυχαίο αριθμό (μικρό) συνιστωσών, θα εκαπιδεύσουμε έναν αλγόριθμο σε ένα τμήμα του συνόλου δεδομένων που έχουμε διαθέσιμο και στη συνέχεια θα μετρίσουμε πόσο αποδοτικός είναι στα υπόλοιπα δεδομένα. Στη συνέχεια θα προσθέσουμε ακόμα μία συνιστώσα θα ξανά εκπαιδεύσουμε τον ίδιο ακριβώς αλγόριθμο και θα ελέξουμε αν υπάρχει αύξηση στα δεδομένα που δεν έχει δει ο

αλγόριθμος. Εάν υπάρξει βελτίωση στην πρόβλεψη της μεταλητής απόκρισης προσθέτουμε άλλη μία συνιστώσα και συνεχίζουμε την ίδια διαδικασία μέχρι να μην υπάρξη αύξηση στην απόδοση του αλγορίθμου. Εάν υπάρξει μείωση στην απόδοση του αλγορίθμου θα αφαιρέσουμε μία κύρια συνιστώσα και θα εφαρμόσουμε την ίδια ακριβώς διαδικασία μέχρις ώτου δεν υπάρξει αύξηση στην απόδοση του αλγορίθμου μας.(θα επεκταθούμε περαιτέρω σε παρακάτω ενότητα).

## 4.2 Singular Value Decomposition

Η Ανάλυση Κύριων Συνιστωσών όπως είδαμε στην ενότητα 3.1 έχει ως στόχο την εύρεση συνιστωσών ασυσχέτιστων μεταξύ τους ώστε να εξηγούν τη μέγιστη δυνατή διακύμανση του συνόλου δεδομένων. Μία άλλη μέθοδος που χρησιμοποιείται για την μείωση των διαστάσεων κάτω από την υπόθεση ότι τα δεδομένα μας μπορούν να αναπαρασταθούν ικανοποιητικά από πίνακα μικρότερης τάξης από ότι ο αρχικός είναι η Singular Value Decomposition.

Χρησιμοποιείται συνήθως στην πράξη σε προβλήματα όπως η αναγνώριση εικόνων όπου η επίλυση του συγκεκριμένου προβλήματος απαιτεί την χρήση πολλών μεταβλητών. Σε αλγόριθμους που προσπαθούν να προβλέψουν τις προτιμήσεις ανθρώπων για παράδεγμα σε ταινίες ανάλογα με το ποιές έχουν δει και πως τις έχουν βαθμολογήσει. Τέλος μιά ακόμα χρήση της συγκεκριμένης τεχνικής είναι στην ομαδοποίηση πελατών μίας επιχείρησης ανάλογα με τα προϊόντα που αγοράζουν. Η συγκεκριμένη μέθοδος έχει ως στόχο δοθέντος ενός πίνακα $A$ να βρει τρείς πίνακες $U, \Sigma, V$ τέτοιους ώστε:

$$A = U\Sigma V^T \tag{4.2.1}$$

όπου ο πίνακας $\Sigma \in R^{n\times n}$είναι διαγώνιος,

ο πίνακας $U \in R^{m\times n}$ είναι ορθοκανονικός

και ο πίνακας $V \in R^{n\times n}$ είναι ορθοκανονικός.

Από την Σχέση (4.2.1) έχουμε ότι :

$$A = \sigma_1 u_1 v_1^T + \sigma_2 u_2 v_1^T + \cdots + \sigma_n u_n v_n^T$$

Εύκολα αποδεικνύεται ότι ο πίνακας $A^T A$ καθώς και ο $AA^T$ είναι συμμετρικοί :

$$(A^T A)^T = A^T A^{T^T} = A^T A$$

$$(\mathrm{AA^T})^\mathrm{T} = A^{T^T} A^T = \mathrm{A}A^T$$

Επίσης εάν $A_{ij} \in R$ τότε οι πίνακες $A^T A, AA^T$ είναι θετικα ημιορισμένοι. Σύμφωνα με το φασματικό θεώρημα υπάρχουν $e_1, e_2, \dots, e_p$ ιδιοδιανύσματα και $\lambda_1, \lambda_2, \dots, \lambda_p$ ιδοτιμές τέτοια ώστε:

$$A^T A = \lambda_1 e_1 e_1' + \lambda_2 e_2 e_2' + \cdots + \lambda_p e_p e_p \tag{4.2.2}$$

Πολλαπλασιάζοντας τον ανάστροφο του πίνακα $A$ με τον πίνακα $A$ έχουμε ότι :

$$A^T A = (U\Sigma V^T)^T U\Sigma V^T$$

$$= (V\Sigma^T U^T) U\Sigma V$$

$$= V\Sigma^\mathrm{T} U^T U\Sigma V^T$$

$$= V\Sigma\Sigma V^T$$

$$= V\Sigma^2 V^T \tag{4.2.3}$$

όπου $UU^T = I$ και $VV^T = I$ αφού είνα ορθογώνιοι

και $\Sigma^T = \Sigma$ αφού είναι διαγώνιος.

Από τις σχέσεις 4.2.2 και 4.2.3 υπολογίζονται οι πίνακες $\Sigma$ και $V$. Για τον υπολογισμό του $U$ από την σχέση 4.2.1 έχουμε ότι :

$$A = U\Sigma V^T$$

$$AV = U\Sigma V^T V$$

$$AV = U\Sigma$$

$$\mathrm{A}V\Sigma^{-1} = U$$

Όπως και στην Ανάλυση Κυρίων Συνιστώσων πρέπει να αποφασιστεί ο βέλτιστος αριθμός singular values που πρέπει να χρησιμοποιηθεί. Η συνολική διακύμανση που ερμηνεύεται από τις singular value ισούται με το άθροισμα των ριζών των ιδιοτιμών του πίνακα $A^T A$. Θα

εφαρμόσουμε τις ίδιες τεχνικές με αυτές που αναφέραμε στην Ανάλυση Κύριων Συνιστωσών ώστε να βρούμε τον βέλτιστο αριθμό singular values.

# 5. Αριθμητική εφαρμογή σε πραγματικά δεδομένα

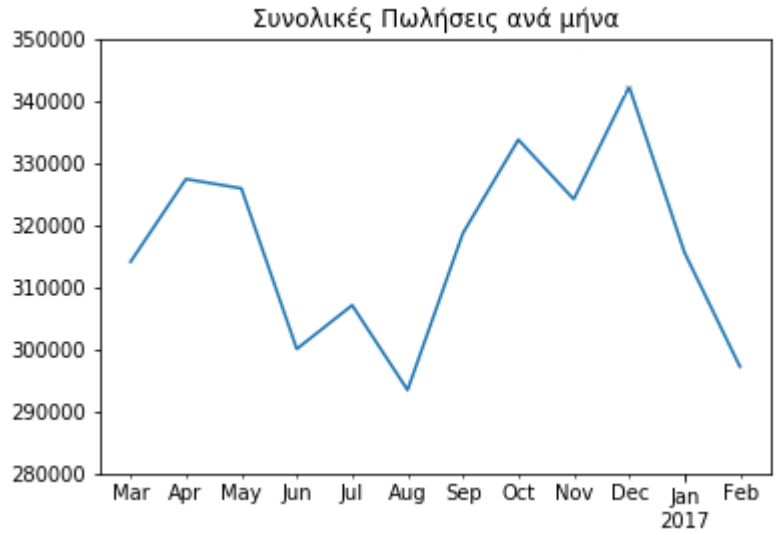
## 5.1 Περιγραφή του συνόλου δεδομένων

Το σύνολο δεδομένων που χρησιμοποιήθηκε αποτελείται από 16 μεταβλητές και 1.876.538 γραμμές. Τα δεδομένα αναφέρονται σε πωλήσεις προϊόντων μίας αλυσίδας σουπερ μάρκετ. Κάθε γραμμή του συνόλου δεδομένων αναφέρεται στην αγορά ενός προϊόντος. Οι στήλες αναφέρονται στο ποιός πελάτης αγόρασε το συγκεκριμένο προϊόν, την ημέρα που το αγόρασε , την ημερομηνία, τον μήνα, την τιμή του προϊόντος, πόσο έκπτωση είχε το συγκεκριμένο προϊόν, από ποιό κατάστημα αγοράστηκε, σε ποιά κατηγορία ανήκει (φαγητά,ρουχα κλπ), σε ποιό ράφι, σε ποίο καλάθι αγοράστηκε το συγκεκριμένο προϊόν και τέλος τον κωδικό του κάθε προϊόντος. Εμπεριόχονται στο συγκεκριμένο σύνολο δεδομένων 932 πελάτες, 204 διαφορετικά καταστήματα, 34460 διαφορετικά προϊόντα και 122567 επισκέψεις πελατών. Τα δεδομένα καλύπτουν την χρονική περίοδο Μάρτιος 2016 εώς Φεβρουάριος 2017. Η μεταβλητή κατηγορία προϊόντος έχει ελλιπείς τιμές συνολικά 67% και συνολικά έχει 495 διαφορετικές κατηγορίες. Οι περισσότερες επισκέψεις γίνονται το Σάββατο. Στον πίνακα 1.1 παρουσιάζονται αναλυτικά οι συνολικές επισκέψεις ανά ημέρα.

| Ημέρα | Επισκέψεις |
|---|---|
| Σάββατο | 413737 |
| Πέμπτη | 312583 |
| Παρασκευή | 308644 |
| Δευτέρα | 249343 |
| Τρίτη | 246694 |
| Τετάρτη | 240199 |
| Κυριακή | 105338 |

Πίνακας 5.1 Σύνολο επισκέψεων ανά ημέρα

Στο Σχήμα 5.1 φαίνονται οι πωλήσεις ανά μήνα και των 204 καταστημάτων.



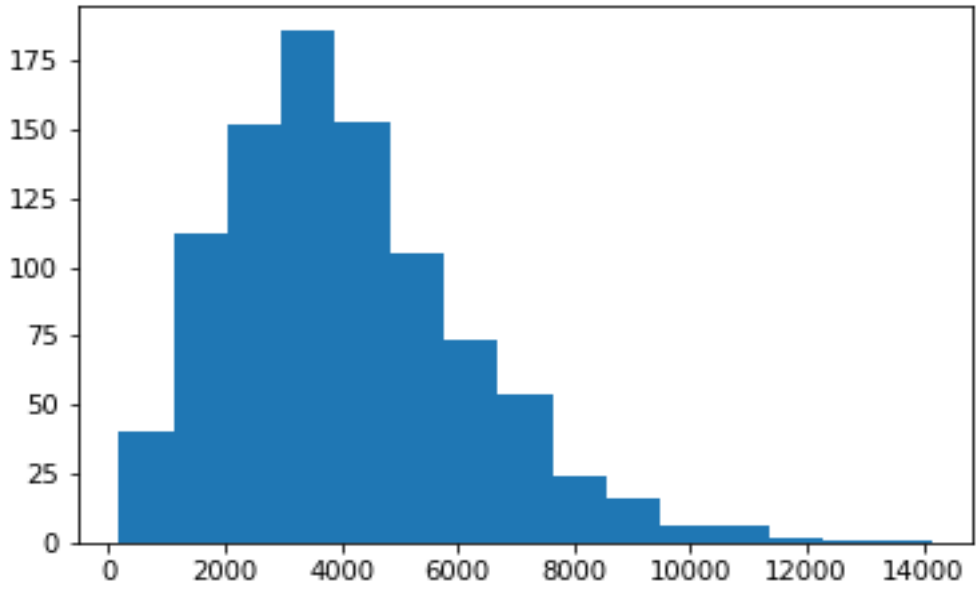

Σχήμα 5.1 Συνολικές πωλήσεις ανά μήνα

Όπως αναμέναμε τους καλοκαιρινούς μήνες είναι πεσμένες οι πωλήσεις, τον Δεκέμβρη και τον Ιανουάριο ανεβασμένες και τον τελευταίο μήνα αρκετά κατεβασμένες. Στον πίνακα 5.2 παρουσιάζονται περιγραφικά στατιστικά του συνολικού ποσού που ξόδεψε κάθε πελάτης.

| | |
|---|---|
| count | 932 |
| mean | 4076.70 |
| std | 2102.88 |
| min | 173.36 |
| 1Q | 2605.75 |
| median | 3748.92 |
| 3Q | 5271.69 |
| max | 14140.79 |

Πίνακας 5.2 Περιγραφικά στατιστικά συνολικού ποσού ανά πελάτη

Στο Σχήμα 5.2 παρουσιάζεται το ιστόγραμμα του συνολικού ποσού ανά πελάτη.




Σχήμα 5.2 Ιστόγραμμα συνολικού ποσού ανά πελάτη.

Από το ιστόγραμμα της μεταβλητής συνολικό ποσό ανά πελάτη παρατηρούμε ότι κατά κύριο λόγο οι πελάτες διαθέτουν συνολικά από 2.000 εώς 6.000 χρηματικές μονάδες με ελάχιστες εξαιρέσεις που ξοδεύουν από 12.000 εώς 14.000. Στην συνέχεια στον πίνακα 5.3 παρουσιάζονται περιγραφικά στατιστικά του μέσου μεταξύ επισκέψεων ανά πελάτη.

| | |
|---|---|
| count | 932 |
| mean | 4.14 |
| std | 3.34 |
| min | 0.63 |
| 1Q | 2.09 |
| median | 3.2 |
| 3Q | 5.16 |
| max | 32.27 |

Πίνακας 5.3 Περιγραφικά στατιστικά του μέσου μεταξύ επισκέψεων ανά πελάτη

Από τον πίνακα 5.3 παρατηρούμε ότι οι πελάτες κατά μέσο όρο ψωνίζουν ανά 4 ημέρες με τον μικρότερο μέσο όρο να είναι 0.6, δηλαδή ο συγκεκριμένος πελάτης επισκέπτεται το κατάστημα και δύο φορές τη μέρα και τη μεγαλύτερη ανά 32 μέρες. Έπειτα στον πίνακα 5.4 αναφέρονται τα περιγραφικά στατιστικά των συνολικών επισκέψεων ανά πελάτη.

| | |
|---|---|
| count | 932 |
| mean | 131.5 |
| std | 84.34 |
| min | 6 |
| 1Q | 68 |
| median | 113 |
| 3Q | 172 |
| max | 571 |

Πίνακας 5.4 Περιγραφικά στατιστικά του συνολικού αριθμού εμφανίσεων ανά πελάτη

Από τον Πίνακα 5.4 παρατηρούμε ότι κατά μέσο όρο οι πελάτες έρχονται 131 φορές στα καταστήματα. Ο μικρότερος αριθμός εμφανίσεων είναι 6 και ο μεγαλύτερος 571, δηλαδή ο συγκεκριμένος πελάτης σε 364 ημέρες έχει επισκευτεί τα καταστήματα 571 φορές. Τέλος στον Πίνακα 5.5 παρουσιάζονται τα περιγραφικά στατιστικά των ημερών που έχουν περάσει από την τελευταία εμφάνιση κάθε πελάτη.

| | |
|---|---|
| count | 932 |
| mean | 3.95 |
| std | 9.95 |
| min | 0 |
| 1Q | 0 |
| median | 1 |
| 3Q | 4 |
| max | 113 |

Πίνακας 5.5 Περιγραφικά στατιστικά των ημερών που έχουν περάσει από την τελευταία εμφάνιση κάθε πελάτη

Παρατηρούμε ότι το 50% των πελατών έκανε την τελευταία του εμφάνιση σε διάστημα λίγοτερο της μίας μέρας, το 75% λιγότερο από 4 ημέρες. Η μεγαλύτερη τιμή είναι αυτή των 113 ημερών.

## 5.2 Ορισμός Προβλήματος

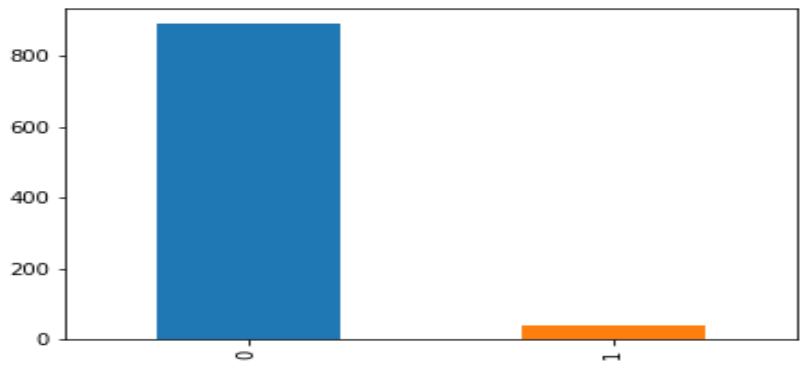
Στη συγκεκριμένη ενότητα γίνεται αναφόρα στο πρόβλημα που επικαλούμαστε να λύσουμε και των τεχνικών που θα χρησιμοποιήσουμε βάση των δεδομένων που έχουμε διαθέσιμα. Το πρώτο πρόβλημα που έχουμε να αντιμετωπίσουμε είναι η πρόβλεψη του αν κάποιος πελάτης θα συνεχίσει να αλλιλεπιδρά με την επιχείρηση ή θα σταματήσει. Το συγκεκριμένο πρόβλημα είναι πολύ δύσκολο καθώς ούτε οι ίδιοι οι πελάτες δεν γνωρίζουν κατά πόσο θα συνεχίσουν να είναι πελάτες σε μία επιχείρηση.

Μία άλλη δυσκολία που έχει το συγκεκριμένο πρόβλημα είναι στο πως ορίζει κανείς την απώλεια ενός πελάτη. Καθώς οι πελάτες δεν έχουν κάποιο συμβόλαιο που με την λήξη και μη ανανέωση του είναι δεδομένη η απώλεια τους. Στη δική μας περίπτωση οι πελάτες αλλιλεπιδρούν με την επιχείρηση ανά πάσα στιγμή. Ουσιαστικά η δυσκολία που αντιμετωπίζουμε είναι στο πως θα χαρακτηρίσουμε ποιοί έχουν φύγει και ποίοι έχουν συνεχίσει να είναι πελάτες, καθώς οι αλγόριθμοι κατηγοριοποίησης που θα εκπαιδεύσουμε προϋποθέτουν την υπάρξη δίτιμης μεταβλητής απόκρισης που διαχωρίζει τις δύο κατηγορίες.

Δοκιμάσαμε διάφορες προσεγγίσεις ώστε να ωρίσουμε την απώλεια πελάτη. Η τελευταία εμφάνιση ενός πελάτη είνα άμεσα συνδεδεμένοι με την απώλεια πελάτη. Έτσι λοιπόν και οι τρείς προσεγγίσεις μας βασίζονται στην τελευταία εμφάνιση. Μία προσεγγίση ήταν να ορίσουμε την απώλεια εώς εκείνων που πέρασε μεγαλύτερο χρονικό διάστημα από τον μέσο όρο μεταξύ των εμφανίσεων τους συν τρεις τυπικές αποκλίσεις από την τελευταία τους εμφάνιση. Στη συνέχεια ορίσαμε την απώλεια εώς εκείνων που πέρασε μεγαλύτερο χρονικό διάστημα από την μέγιστη ημέρα μεταξύ των εμφανίσεων τους από την τελευταία τους εμφάνιση. Τα καλύτερα αποτελέσματα απέφερε η τελευταία μέθοδος που όρισε την απώλεια εώς εκείνων που πέρασαν πάνω από 14 ημέρες από την τελευταία τους εμφάνιση.

Στη συνέχεια το άλλο πρόβλημα που έχουμε να λύσουμε είναι η συνολική αξία πελάτη (CLV). Ουσιαστικά καλούμαστε να προβλέψουμε πόσα θα ξοδέψει ένας πελάτης μελλοντικά στην επιχείρηση. Όπως και η απώλεια πελάτη έτσι και η συνολική του αξία για την επιχείρηση δεν είναι εύκολο πρόβλημα. Το να προσπαθήσει κανείς να προβλέψει την συνολική αξία κάθε πελάτη σε χρονικό ορίζοντα ενός χρόνου φαντάζει ακατόρθωτο. Για το λόγο αυτό επιλέξαμε χρονικό ορίζοντα ενός και δύο μηνών.

Όλες οι επεξηγηματικές μεταβλητές που χρησιμοποιήθηκαν για την εκπαίδευση των αλγορίθμων που περιγράψαμε στην Ενότητα 2 πάρθηκαν το χρονικό διάστημα Μάρτιος 2016 εώς Ιανουάριος 2017 για την συνολική αξία πελάτη ενός μήνα. Οι μεταβλητές απόκρισεις πάρθηκαν από το Φεβρουάριο 2017 για τον ένα μήνα και αντίστοιχα για τους δύο μήνες από το χρονικό διάστημα Μάρτιος 2016 εώς Δεκέμβριος 2016 οι επεξηγηματικές και από τον Ιανουάριο, Φεβρουαάριο 2017 οι μεταβλητές απόκρισης. Στη συνέχεια οι προβλέψεις μας έγιναν χρησιμοποιώντας επεξηγηματικές μεταβλητές από το χρονικό διάστημα Απρίλιος 2016 εώς Φεβρουάριος 2017 για την πρόβλεψη ενός μηνός και Μάϊος 2016 εως Φεβρουάριος 2017 γία δύο μήνες. Ο λόγος που έγινε ο συγκεκριμένος διαχωρισμός μεταξύ των επεξηγηματικών και μεταβλητων απόκρισεις ήταν ώστε να μην δίνεται η πληροφορία στο μοντέλο μας σχετικά με το πότε ήταν η τελευταία εμφάνιση του κάθε πελάτη και πόσα ξόδεψε τον μήνα που θέλουμε να προβλέψει. Περισσότερες πληροφορίες σχετικά με τις μεταβλητές που δημιουργήθηκαν και τους μετασχηματισμούς που έγιναν δίνονται στην επόμενη παράγραφο. Στο Σχήμα 5.3 φαίνεται ο αριθμός όσων παρέμειναν να είναι πελάτες και όσοον έφυγαν από την επιχείρηση. Για τον καθορισμό τον απωλεσθέντων χρησιμοποιήθηκε το κριτήριο των 14 μερών που ανεφέρθηκε νωρίτερα.



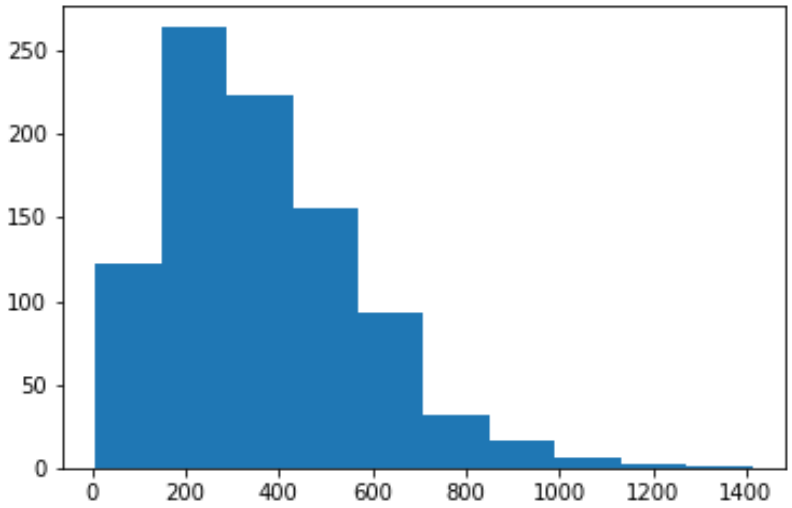

Σχήμα 5.3 Αριθμός απωλεσθέντων έναντι μη απωλεσθέντων

Από το Σχήμα 5.3 παρατηρούμε ότι ο αριθμός απωλεσθέντων είναι σημαντικά μικρότερος από ότι ο αριθμός όσων παραμένουν ακόμα πελάτες. Τα περισσότερα πραγματικά προβλήματα κατηγοριοποίησης αποτελούνται από μεταβλητές απόκρισης άνισου μεγέθους μεταξύ των ομάδων και κατά κύριο λόγο η ομάδα με τις λιγότερες παρατηρήσεις είναι αυτή που μας ενδιαφέρει. Για παράδειγμα στο δικό μας πρόβλημα ζωτικής σημασίας αποτελεί η ορθή πρόβλεψη των πελατών που θα σταματήσουν να αλλιλεπιδρούν με την επιχείρηση, οι οποίοι αποτελούν και την μειοψηφική ομάδα.

Οι αλγόριθμοι που θα εκπαιδεύσουμε θα αξιολογηθούν με το κριτήριο AUC το οποίο είναι το εμβαδόν κάτω από την καμπύλη Receiver Operating Characteristic (ROC). Η συγκεκριμένη καμπύλη δημιουργείται απεικονίζοντας το πσοστό των αληθώς θετικών έναντι των ψευδών θετικών που έχει εκτιμήσει ένα μοντέλο σε όλα τα δυνατά κατώφλια, όπου σαν κατώφλι ορίζεται η τιμή εκείνη που κοθορίζει σε ποιά ομάδα θα ταξινομηθεί κάθε παρατήρηση ανάλογα με την εκτιμώμενη πιθανότητα. Πιθανότητες με μικρότερη τιμή από το κατώφλι ταξινομούνται στην πρώτη ομάδα και πιθανότητας με μεγαλύτερη στην δεύτερη. Η περιοχή κατώ από την καμπύλη ROC (AUC) είναι ουσιαστικά η πιθανότητα ένα μοντέλο να εκτιμήσει υψηλότερη πιθανότητα σε μία παρατήρηση που ανήκει στην πρώτη ομάδα από ότι μία που ανήκει στην δεύτερη ομάδα. Οι τιμές που μπορεί να παρεί η AUC είναι από 0 εώς 1. Τιμές κοντά στο 0.5 αξιολογούν την ικανότητα του μοντέλου μας ως μη αποδοτική, αντιθέτως τιμές κοντά στο 0 και στο 1 ως σημαντικά αποδοτική.

Ένα άλλο πρόβλημα που δημιουργεί η ανισότητα των ομάδων της μεταβλητής απόκρισης είναι στην εκπαίδευση των αλγορίθμων, καθώς όπως γνωρίζουμε η εκπαίδευση τους γίνεται μέσα από παραδείγματα. Δίνοντας σε ένα αλγόριθμο όπως στην δική μας περίπτωση 890 παρατηρήσεις που ανήκουν στην πρώτη ομάδα και μόνο 40 στην δεύτερη οδηγεί συνήθως στην μεροληψία ως προς την πρώτη ομάδα. Δοκιμάστηκαν δύο τεχνικές για την αντιμετώπιση του προβλήματος όπως υπερδειγματοληψία της μειονεκτούσας ομάδας μέσα από την δημιουργία συνθετικών παρατηρήσεων και εισαγωγή βαρών στην συνάρτηση ελαχιστοποίησης ώστε να τιμωρείται περισσότερο η λανθασμένη ταξινόμηση παρατήρησης που ανήκει στην μειονεκτούσα ομάδα. Και οι δύο τεχνικές συντέλεσαν στην ελάχιστη αύξηση του AUC. Το πρόβλημα που δημιούργησαν ήταν στην εκτίμηση των πιθανοτήτων, καθώς αλλάζοντας τις prior πιθανότητες με την υπερδειγματολειψία και προσθέτοντας βάρη στην συνάρτηση κόστους το μοντέλο μας ταξινομούσε σωστά τις πιθανότητες αλλά υπερεκτιμούσε ότι κάποιος θα φύγει. Στο πρόβλημα που επικαλούμαστε να λύσουμε απαιτείται η σωστή εκτίμη της

πιθανότητας απώλειας πελάτη και όχι τοσο ο σωστός διαχωρισμός μεταξύ των ομάδων (θα αναφερθούμε περαιτέρω σε επόμενη ενότητα). Στο Σχήμα 5.4 απεικονίζεται η κατανομή της μεταβλητής απόκρισης συνολική αξία πελάτη (CLV) ενός μήνα.




Σχήμα 5.4 Ιστόγραμμα της μεταβλητής απόρκρισης CLV ενός μήνα

Όπως αναμέναμε η κατανομή έχει περίπου ίδιο σχήμα με εκείνη του συνολικου ποσού ανά πελάτη(βλ. Σχήμα 5.2). Στον πίνακα 5.6 παρουσιάζονται ορισμένα περιγραφικά στατιστικά της μεταβλητής απόκρισης CLV ενός μήνα.

| | |
|---|---|
| count | 932 |
| mean | 359.35 |
| std | 213.61 |
| min | 0 |
| 1Q | 201.52 |
| median | 327.10 |
| 3Q | 496.92 |
| max | 1412.67 |

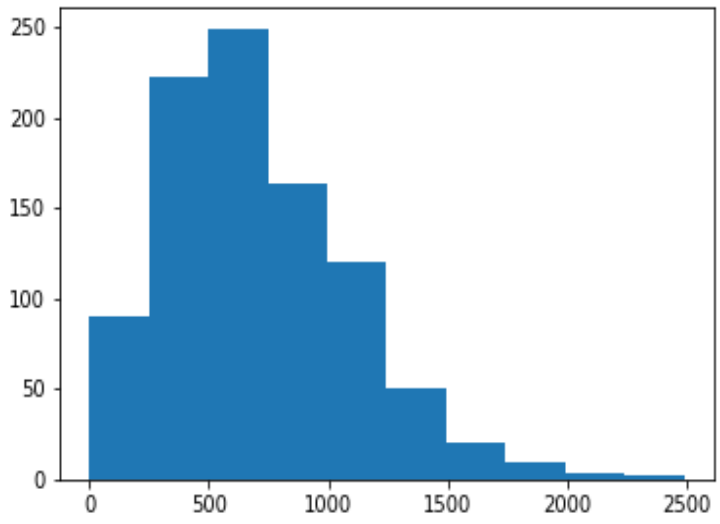
Πίνακας 5.6 Περιγραφικά στατιστικά της μεταβλητής απόκρισης CLV ενός μήνα

Από τον πίνακα 5.6 παρατηρούμε ότι η μέση τιμή της μεταβλητής απόκρισης ισούται με 359 και το τρίτο τεταρτημόριο ισούται με 486, δηλαδή το 75% των πελατών ψωνίζουν μέχρι 486 ευρώ. Υπολογίζοντας το 0.9 ποσοστιμόριο ισούται με 632, γεγονός που καθιστά την πρόβλεψη των πελατών υψηλής αξίας μεγαλύτερη των 800 χρηματικών μονάδων ιδαιέτερα δύσκολη καθώς οι παρατηρήσεις με CLV μεγαλύτερη του 800 είναι ελάχιστες. Για την αντιμετώπιση του συγκεκριμένου προβλήματος θα αναφερθούμε σε επόμενη ενότητα. Για την αξιποιστία του μοντέλου μας θα αποφανθούμε βάσει του κριτηρίου της ρίζα του μέσου τετραγωνιού σφάλματος.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{N}}$$

όπου $y_i$ η πραγματική τιμή και $\hat{y}_i$ η προβλεπόμενη τιμή.

Τέλος στο Σχήμα 5.5 παρουσιάζεται και η κατανομή της CLV δύο μηνών.




Σχήμα 5.5 Ιστόγραμμα της μεταβλητής απόκρισης CLV δύο μηνών

Η κατανομή της CLV είναι σχεδόν παρόμοια για ένα και δύο μήνες με την μόνη διαφορά να είναι στο πεδίο τιμών τους. Στον Πίνακα 5.7 παρουσιάζονται βασικά περιγραφικά στατιστικά της CLV δύο μηνών.

| | |
|---|---|
| count | 932 |
| mean | 706.64 |
| std | 393.17 |
| min | 0 |
| 1Q | 432.19 |
| median | 647.42 |
| 3Q | 942.72 |
| max | 2486.95 |

Πίνακας 5.7 Περιγραφικά στατιστικά της μεταβλητής απόκρισης CLV δύο μηνών

## 5.3 Προκαταρτική επεξεργασία δεδομένων

Η εξαγωγή, επιλογή και ο μετασχηματισμός δεδομένων αποτελεί σημαντικό μέρος μίας στατιστική ανάλυση. Σε πραγματικά προβλήματα όπως το δικό μας ο ερευνητής δεν έχει στη διάθεση του έτοιμες μεταβλητές και πρέπει να τις εξάγει μέσα από τα δεδομένα. Στην παράγραφο αυτή θα αναλύσουμε μεταβλητές που κατασκευάσαμε για την επίλυση των δύο προβλημάτων που αναφέραμε στην προηγούμενη παράγραφο καθώς και στους μετασχηματισμούς που προχωρήσαμε ώστε να εφαρμοστούν στους αλγορίθμους που περιγράψαμε στην δεύτερη ενότητα.

Αρχικά έπρεπε να αναλογιστούμε τα δύο προβλήματα που έχουμε να λύσουμε ώστε να κατασκεύασουμε μεταβλητές που θα μας βοηθήσουν στην επίλυσή τους. Κατασευάσαμε 42 μεταβλητές βασιμέμενες στις μεταξύ επισκέψεις των πελατών και το καλάθι που ψώνιζαν σε κάθε επίσκεψη. Για τις μεταξύ επισκέψεις δημιουργήθηκαν μεταβλητές όπως ο μέσος όρος, η διάμεσος, η τυπική απόκλιση, μέγιστη ελάχιστη επικρατούσα τιμή, η κύρτωση και ο συντελέστής ασυμετρίας. Ακόμη δημουργήθηκαν μεταβλητές όπως οσυνολικός αριθμός επισκέψεων, πόσες ημέρες πέρασαν ώστε να πραγματοποιήσει ένας πελάτης επίσκεψη, πόσες

πέρασαν από την τελευταία του επίσκεψη και οι ημέρες που ένας πελάτης ήταν ενεργός (τελευταία μείον αρχική εμφάνιση).

Για το καλάθι του κάθε πελάτη δημιουργήθηκαν οι ίδιες περιγραφικές μεταβλητές με τις μεταξύ επισκέψεις πελατών (μέσος όρος, τυπική απόκλιση κλπ). Επίσης κατασκευάστηκαν το συνολικό ποσό και το σύνολο των διαφορετικών προϊόντων που αγοράστηκε ανά πελάτη. Ακόμη κατασκευάστηκαν μεταβλητές βασισμένες στην εντροπία όπως η ημέρα που επιλέγει κάποιος να πραγματοποιήσει την επισκεψή του, το κατάστημα που επιλέγει και τέλος το πόσο που διέθεται μεταξύ των μηνών.Στην μεταβλητή ημέρα για παράδειγμα μεγάλη εντροπία υποδηλώνει ότι κάποιος επισκέπτεται το κατάστημα διαφορετικές ημέρες ενώ μικρή ότι το επισκέπτεται συγκεκριμένες και μηδενικη τιμή ότι το επισκέπτεται μία συγκεκριμένη μέρα. Δημιουργήθηκε και μία μεταβλητή αλλιλεπίδρασης που ορίζεται εώς ο λόγος μεταξύ της τελευταίας επίσκεψης και στον μέσο όρο επισκέψεων ανά πελάτη.

Τέλος επειδή το πρόβλημα που έχουμε να λύσουμε είναι χρονολογικό προστέθηκαν και χρονολογικές μεταβλητές όπως ο μέσος όρος των τελευταίων τριών και πέντε μεταξύ επισκέψεων. Οι συνολικές επισκέψεις ανά πελάτην τον τελευταίο, προτελευταίο και τρεις μήνες πριν. Η πρόβλεψη της εκθετικής εξομάλυνσης για $\alpha = 0.9,0.5$. Για το καλάθι δημιουργήθηκαν οι ίδιες χρονολογικές μεταβλητές.

Αφου ολοκληρώθηκε το στάδιο της εξαγωγής μεταβλητών από το σύνολο των δεδομένων στη συνέχεια εργαστίκαμε στην επιλογή και απόρριψη εκείνων που ενδεχομένως να μην μπορούν να βοηθήσουν στην πρόβλεψη της απώλειας και συνολικής αξίας πελάτη. Αρχικά για την απώλεια πελάτη απορρίψαμε 16 μεταβλητές έχοντας σαν κριτήριο τον συντελεστή του Pearson. Ουσιαστικά ορίσαμε ένα κατώτερο κατώφλι 0.10 και όσες μεταβλητές είχαν λιγότερο αφαιρέθηκαν. Στον Πίνακα 5.8 φαίνονται 10 μεταβλητές με τον μεγαλύτερο συντελεστή συσχέτισης τους με την μεταβλητή απόκρισης ταξινομημένες κατά απόλυτη τιμή σε φθίνουσα σειρά.

| churn | correlation |
|---|---|
| last_purchase | 0.373450 |
| ma_5 | 0.29689 |
| std_btv | 0.289636 |
| purchase_mean | 0.280232 |
| ma_3 | 0.268914 |
| mean_btv | 0.266158 |
| month_entr | 0.262367 |
| ewma_0.5 | 0.260517 |
| active_days | 0.243235 |
| amount_lag1 | 0.227554 |

Πίνακας 5.8 Συντελεστής συσχέτισης μεταξύ μεταβλητής απόκρισης και επηξηγηματικών

Όπως αναμέναμε η μεταβλητή ημέρες που πέρασαν από την τελευταία του αγορά (last_purchase) έχει την υψηλότερη συσχέτιση με την μεταβλητή απόκρισης. Η μεταβλητή μέσος όρος των 5 τελευταιίων χρόνων μεταξύ επισκέψεων (ma_5) είναι η αμέσως επόμενη υψηλότερη σε συσχέτιση με την απόκριση, γεγονός αναμενόμενο καθώς μεταβλητές που αφορούν τις τελευταίες ενέργειες ενός πελάτη συνήθως καθορίζουν την απώλεια του ή οχί. Μία τέτοια είναι και το ποσό που διέθεσε τον τελευταίο μήνα (amount_lag1). Μία άλλη ενδιαφέρουσα μεταβλητή είναι η τυπική απόκλιση του χρόνου μεταξυ των επισκέψεων πελάτη, που υποδηλώνει ότι πελάτες που τείνουν να επισκέπτονται καταστήματα έχοντας μεγάλη μεταβλητότητα στον χρόνο μεταξύ των επισκέψεων τους είναι πολύ πιθανό να οδηγήσει στην απώλεια τους. Τέλος η αλλιλεπίδραση μεταξύ του χρόνου από την τελευταία εμφάνιση και του μέσου χρόνου (purchase_mean) μεταξύ εμφανίσεων έχει σχετικά υψηλή συσχέτιση με την απώλεια ενός πελάτη.

Στη συνέχεια εκπαιδεύουμε έναν αλγόριθμο κατηγοριοποίησης Random Forest ώστε να πάρουμε μία γενική εικόνα για την σημαντικότητα των μεταβλητών με σκοπό την πρόβλεψη απώλειας ενός πελάτη. Ο συγκεκριμένος αλγόριθμος αξιολογεί την σημαντικότητα των μεταβλητών ως αυτές που συντέλεσαν περισσότερο στην δημιουργία κόμβων, δηλαδή εκείνες που διαχωρίζοντας το σύνολο τιμών τους αποφέρεται η μεγαλύτερη μείωση στη συνάρτηση ελαχιστοποίησης που έχει οριστεί από τον χρήστη. Μεταβλητές που βρίσκονται στο πρώτο επίπεδο ενός δέντρου χαρακτηρίζονται σημαντικότερες από ότι εκείνες που βρίσκονται σε πιο

κάτω επίπεδα. Στον Πινακα 5.9 παρουσιάζονται οι 10 σημαντικότερες μεταβλητές ταξινομημένες κατά φθίνουσα σείρα.

| churn | Σημαντικότητα |
|---|---|
| purchase_mean | 0.060232 |
| amount_lag1 | 0.057554 |
| last_purchase | 0.047345 |
| events_lag1 | 0.0469 |
| ewma_0.5 | 0.026051 |
| active_days | 0.0243235 |
| std_btv | 0.0219636 |
| ma_3 | 0.0206889 |
| amount_lag3 | 0.009638 |
| events | 0.0096209 |

Πίνακας 5.9 Σημαντικότητα μεταβλητών για την πρόβλεψη απώλειας πελάτη

Η συγκεκριμένη τεχνική δεν είναι 100% αντιπροσωπευτηκεί καθώς πολλές μεταβλητές που έχουμε στο μοντέλο μας είναι ηψηλά συσχετισμένες μεταξύ τους, οδηγώντας συνήθως σε υποεκτίμηση ορισμένων που είναι ηψηλά συσχετισμένες με άλλες και είναι ελάχιστα σημαντικότερες από ότι αυτές. Την λύση στο πρόβλημα αυτό δίνει η ανάλυση κύριων συνιστωσών που προβάλει το σύνολο των δεδομένων σε συνιστώσες ασυσχέτιστες μεταξύ τους λιγότερων διαστάσεων από ότι οι αρχικές. Στον Πίνακα 5.10 παρουσιάζονται οι 10 μεγαλύτερες συσχετίσεις μεταξύ των επεξηγηματικών μεταβλητών όπως αυτές προέκυψαν από τον συντελεστή γραμμικής συσχέτισης του Pearson.

| | | |
|---|---|---|
| min_basket | mode_basket | 0.990217 |
| mean_basket | median_basket | 0.975734 |
| ewma_basket0.5 | ma_basket3 | 0.957395 |
| ewma_0.5 | ma_3 | 0.939402 |
| active_days | first_purchase | 0.938803 |
| mean_btv | median_btv | 0.932150 |
| ma_basket3 | ma_basket5 | 0.928244 |
| ewma_0.9 | ewma_0.5 | 0.928176 |
| events | events_lag3 | 0.927683 |
| ewma_basket0.9 | ewma_basket0.5 | 0.927542 |

Πίνακας 5.10 Συντελεστής συσχέτισης μεταξύ επεξηγηματικών μεταβλητών

Παρατηρούμε ότι οι περισσότερες επεξηγηματικές μεταβλητές είναι συσχετισμένες μεταξύ τους. Που σημαίνει ότι δεν μπορούμε να τις εισάγουμε όλες στον αλγόριθμο που θα εκπαιδεύσουμε καθώς έχοντας όλες τις μεταβλητές στο μοντέλο θα οδηγηθούμε σε υπερπροσαρμογή του στο σύνολο δεδομένων εκπαίδευσης, κάνοντας το μη αποτελεσματικό σε νέα σύνολα δεδομένων.

Στη συνέχεια εφαρμόζουμε την τεχνική ανάλυση κυρίων συνιστωσών με σκοπό την μείωση και δημιουργία ασυσχέτιστων νέων μεταβλητών. Όπως αναφέραμε σε προηγούμενη ενότητα το πρόβλημα που τείθεται να λύσει κανείς εφαρμόζοντας την συγκεκριμένη τεχνική είναι ο βέλτιστος αριθμός κύριων συνιστωσών. Αναλύσαμε διάφορες προσεγγίσεις για την αντιμετώπιση του συγκεκριμένου προβλήματος καταλήγοντας στην χρήση ενός δείγματος του συνόλου των δεδομένων για να εκπαιδεύσουμε τον αλγόριθμο και άλλο ένα γιά να αποφανθούμε για τον βέλτιστο αριθμό συνιστωσών που θα χρησιμοποιήσουμε.

Αρχικά χωρίζουμε το σύνολο των δεδομένων σε τρία υποσύνολα τυχαία. Έχοντας σαν κριτήριο και τα τρία υποσύνολα να έχουν κοινό ποσοστό της μειοψηφούσας ομάδας, με σκοπό να διασφαλίσουμε την ύπαρξη παρατηρήσεων και των δύο ομάδων σε όλα τα υποσύνολα. Το πρώτο σύνολο, το οποίο διαθέτει το 40% των παρατηρήσεων,θα χρησιμοποιηθεί για την εκπαίδευση των αλγορίθμων (σύνολο εκπαίδευσης). Το δεύτερο, το οποίο διαθέτει το 30\%

των παρατηρήσεων, για να βρούμε τον βέλτιστο αριθμό κύριων συνιστωσών και των παραμέτρων που χρειάζεται να ορίσουμε σε κάθε αλγόριθμο (σύνολο επικύρωσης). Το τρίτο, το οποίο διαθέτει το 30\% των παρατηρήσεων, για να ελέξουμε την αποτελεσματικότητα του τελικού μας μοντέλου σε δεδομένα που δεν έχει ξανά δει (σύνολο ελέγχου). Έπειτα κανονικοποιούμε τα τρία υποσύνολα που δημιουργήσαμε καθώς η συγκεκριμένη τεχνική είναι ευαίσθητη σε μεταβλητές διαφορετικών μονάδεων μέτρησης. Κανονικοποιώντας τα υποσύνολα εξασφαλίζουμε ότι όλες οι μεταβλήτες μας έχουν κοινή μέση τιμή 0 και τυπική απόκλιση 1.

Ξεκινήσαμε με σχετικά μικρό αριθμό συνιστωσών εαν σκεφτεί κανείς ότι το αρχικό σύνολο δεδομένων είχε 26 μεταβλητές. Εκπαιδεύσαμε 4 αλγορίθμους ένα Random Forest, ένα XGBoost, ένα LightGBM και μία λογιστική παλινδρόμηση. Στη συνέχεια κοιτάζουμε την απόδοση του αλγορίθμου μας στο δεύτερο σύνολο. Το μέτρο που χρησιμοποιούμε για να μετρήσουμε την αποτελεσματικότητα κάθε αλγορίθμου είναι το εμβαδόν κατώ από την καμπύλη Receiver Operating characteristic (AUC). Τα καλύτερα αποτελέσματα επέφερε η χρήση 8 κύριων συνιστωσών. Με τη χρήση 8 κύριων συνιστωσών ερμηνεύεται το 90% της αρχικής μεταβλητότας των μεταβλητών μας, τις οποίες θα χρησιμοποιήσουμε για την πρόβλεψη της απώλειας πελάτη. Όσον αφορά την πρόβλεψη αξία συνολικής αξιάς πελάτη δεν θα χρησιμοποιηθούν 17 μεταβλητές έχοντας σαν κριτήριο τον συντελεστή συσχέτισης του Pearson και κατώτερο κατώφλι αυτή τη φορα το 0.2. Στον πίνακα 5.11 ταξινομούνται οι επεξηγηματικές μεταβλητές ανάλογα με την κατά απόλυτη τιμή του συντελεστή συσχέτισης Pearson με την επεξηγηματική μεταβλητή συνολική αξία πελάτη.

| CLV | correlation |
|---|---|
| amount | 0.838394 |
| amount_lag1 | 0.819779 |
| amount_lag3 | 0.797827 |
| amount_lag2 | 0.79417 |
| diff_products | 0.682369 |
| events_lag1 | 0.517547 |
| events | 0.508473 |
| events_lag2 | 0.506959 |
| events_lag3 | 0.505310 |
| mean_btv | 0.454300 |

Πίνακας 5.11 Συντελεστής συσχέτισης μεταξύ μεταβλητής απόκρισης και επεξηγηματικών

Όπως αναμέναμε οι μεταβλητές που υποδηλώνουν πόσα έχει ξοδέψει ένας πελάτης συνολικά, τον τελευταίο μήνα, τον πρότελευταίο και τρεις μήνες πριν είνα αρκετά συσχετισμένες με την CLV. Ο αριθμός των διαφορετικών προϊόντων που έχει αγοράσει κάποιος επίσης έχει υψηλή συσχέτιση με το πόσα θα ξοδέψει στο μέλλον. Επίσης ο αριθμός των επσκέψεων που πραγματοποίησε τον τελευταίο μήνα, συνολικά, πρίν δύο μήνες και πριν τρείς όπως φαίνεται συντελούν στην πρόβλεψη της αξίας πελάτη. Όπως και στην πρόβλεψη απώλειας πελάτη ετσί και τώρα θα εκπαιδεύσουμε έναν αλγόριθμο παλινδρόμησης Random Forest με σκοπό να δούμε ποίες μεταβλητές βοήθησαν περισσότερο στην εκπαίδευση του αλγορίθμου. Στον πίνακα 5.12 παρουσιάζονται οι 10 πιο σημαντικές μεταβλητές που βοηθούν στην πρόβλεψη της μεταβλητής απόκρισης CLV.

| CLV | Σημαντικότητα |
|---|---|
| amount | 0.547222 |
| amount_lag1 | 0.185109 |
| amount_lag2 | 0.042244 |
| amount_lag3 | 0.041012 |
| probs | 0.014460 |
| purchase_mean | 0.010201 |
| month_entr | 0.008436 |
| diff_products | 0.008425 |
| std_btv | 0.007873 |

Πίνακας 5.12 Σημαντικότητα μεταβλητών για την πρόβλεψη της συνολικής αξίας πελάτη

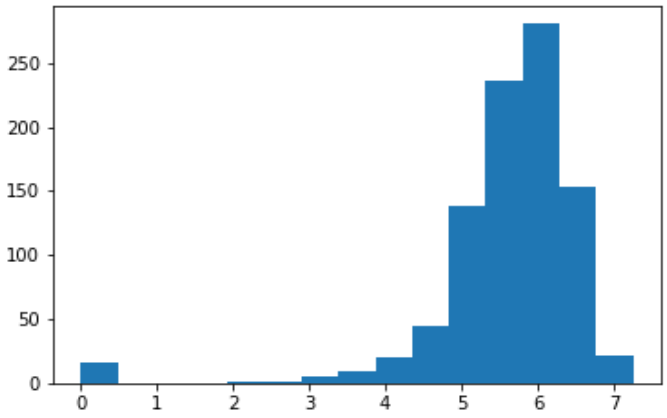
Από τον πίνακα 5.12 παρατηρούμε ότι καθοριστικό ρόλο στην πρόβλεψη συνολικής αξίας πελάτη συντελούν οι μεταβλητές που υποδηλώνουν πόσα έχει ξοδέψει ένας πελάτης συνολικά, τον τελευταίο μήνα, τον πρότελευταίο και τρεις μήνες πριν. Επίσης οι πιθανότητες που εκτιμήσαμε για τον ρυθμό απώλειας πελάτη φαίνεται να είναι σημαντικές για την πρόβλεψη της απόκρισης. Οι υπόλοιπες μεταβλητές φαίνεται να βοηθούν ελάχιστα την πρόβλεψη της συνολικής αξίας πελάτη χωρίς βέβαια να μπορούν να χαρακτηριστούν ασήμαντες. Τέλος επισημαίνεται πως η συγκεκριμένη τεχνική για τους λόγους που προαναφέρθηκαν δεν είναι 100% αντιπροσωπευτηκεί και η χρήση της γίνεται με σκοπό να πάρουμε μία γενική εικόνα για την σημαντικότητα των μεταβλητών μας.

Όπως και στην πρόβλεψη της απώλειας έτσι και στην συνολική αξία πελάτη χρησιμοποιείται η τεχνική ανάλυση κύριων συνιστωσών καθώς οι μεταβλητές μας έχουν πολύ υψηλή συσχέτιση. Το πρόβλημα που δημιουργείται είναι πάλι ο υπολογισμός του βέλτιστου αριθμού συνιστωσών. Η προσέγγιση που ακολουθείται είναι ακριβώς η ίδια με τη μόνη διαφορά ότι εκπαιδεύουμε αλγόριθμους παλινδρόμησης. Το κριτήριο που χρησιμοποιείται για την μέτρηση της αποδοτικότητας των αλγορίθμων είναι η ρίζα του μέσου τετραγωνικού σφάλματος RMSE. Οι αλγόριθμοι που εκπαίδευσαμε ήταν οι XGBoost, LightGBM, Random Forest, Γραμμική Παλιδρόμηση. Ο βέλιστος αριθμός κύριων συνιστωσών βρέθηκε 8, οι οποίες μαζί ερμηνεύουν

88% της αρχικής μεταβλητότητας του συνόλου των δεδομένων μας. Στην μεταβλητή απόκρισης CLV πραγματοποιήθηκε ο εξής μετασχηματισμός:

$$Y' = log(Y + 1)$$

ο οποίος έδωσε στην μεταβλητή απόκρισης ένα πιο καμπανωείδες σχήμα και απέφερε καλύτερα αποτελέσματα. Στο Σχήμα 5.6 φαίνεται η κατανόμη της CLV ενός μήνα μετά τον μετασχηματισμό της.




Σχήμα 5.6 Ιστόγραμμα της μεταβλητής απόκρισης CLV μετά τον μετασχηματισμό της.

Τέλος όσες παρατηρήσεις είχαν αριθμό εμφανίσεων λιγότερο των 10 ημερών διαγράφηκαν από το σύνολο δεδομένων εκπαίδευσης καθώς χαρακτηρίστηκαν ως θορυβόδεις. Έχοντας παρατηρήσεις με λιγότερες από 10 εμφανίσεις μέσα σε ένα χρόνο σε καταστήματα σούπερ μάρκετ δεν μπορούν να χαρακτηριστούν ως πελάτες της αλυσίδας. Συμπεριλαμβάνοντας τις συγκεκριμένες παρατηρήσεις εκαπιδεύουμε αλγορίθμους σε παρατηρήσεις που ουσιαστικά δεν είναι πελάτες της επιχείρησης το οποίο και δεν επιθυμούμε.

## 5.4 Ερμηνεία Προϊόντων

Στην προηγούμενη ενότητα αναφερθήκαμε στην εξαγωγή μεταβλητών που είχαν σχέση με την συχνότητα και το καλάθι της κάθε εφμφάνισης ανά πελάτη. Δεν έγινε καμία αναφορά στα

προϊόντα που αγοράστηκαν ανά πελάτη εκτός από μία μεταβλητή που αναφερόταν στο σύνολο τους. Όπως προαναφέρθηκε στο σύνολο δεδομένων εμπεριέχονται 34.460 διαφορετικά προϊοντα η εισαγωγή τους στο μοντέλο μας είναι ανέφικτη καθώς δεν έχουμε διαθέσιμους τόσους βαθμούς ελευθερίας αλλά και να είχαμε παλί δεν θα οδηγούσε η εισαγωγή τους σε ικανοποιητικά αποτελέσματα. Οι άλλες δύο μεταβλητές που αφορούν προϊόντα είναι η κατηγορία στην οποία ανήκει το κάθε ένα καθώς και σε ποιό ράφι με την πρώτη να έχει 495 διαφορετικές κατηγορίες και τη δεύτερη 122. Πάλι για τους ίδιους λόγους η εισαγωγή τους στο μοντέλο δεν συνιστάται. Ένας άλλος λόγος είναι ότι η κατηγορία του προϊόντος και το ράφι στο οποίο ανήκουν δεν αποφέρει ικανοποιητική πληροφορία για την αξία του προϊόντος ώστε μέσα από αυτήν να εξάγουμε συμπεράσματα για την CLV. Σε αντίθεση ένα ακριβό προϊόν που έχει αγοραστεί από συγκεκριμένους πελάτες μπορεί να μας προδιαθέσει ότι οι συγκεκριμένοι πελάτες έχουν υψηλή συνολική αξία. Το ερώτημα που τίθεται είναι ποιά τενική θα εφαρμοστεί ώστε να αξιοποιήσουμε αυτή την πληροφορία. Εξερευνόντας το σύνολο των δεδομένων μας ως προς την αγοραστική συχνότητα των προϊόντων παρατηρούμε ότι ένα 10% των προϊόντων μας αγοράστηκε πάνω από 100 φορές. Στον Πίνακα 5.13 παρουσιάζονται βσικά περιγραφικά στατιστικά της μεταβλητής αγοραστικές συχνότητες προϊόντων.

| | |
|---|---|
| count | 34360 |
| mean | 54 |
| std | 232 |
| min | 1 |
| 1Q | 1 |
| median | 4 |
| 3Q | 26 |
| max | 17039 |

Πίνακας 5.13 Περιγραφικά στατιστικά προϊόντων

Θα χρησιμοποιήσουμε όσα προϊόντα έχουν αγοραστεί πάνω από 100 φορές ώστε να μπορέσουμε να βρούμε κάποια αλλιλεπίδραση μεταξύ πελατών στις αγοραστικές τους συνήθειες σε προϊόντα χαμηλού και ακριβού κόστους. Αφαιρώντας όσα προϊόντα έιχαν αγοραστεί λιγότερο από 100 φορές καταλήγουμε σε 3801 προϊόντα τα οποία και πάλι είναι πάρα πολλά για να χρησιμοποιηθούν όλα. Τη λύση στο πρόβλημα δίνει η τεχνική Singular

Value Decomposition που περιγράψαμε σε προηγούμενη ενότητα. Ο πίνακας Α στην συγκεκριμένη περίπτωση έχει 932 γραμμές και 3801 στήλες. Θέλουμε να μειώσουμε τις διαστάσεις του μέσω της τεχνικής Singular Value Decomposition και να βρούμε ένα βέλτιστο αριθμό singular values με σκοπό την αύξηση του προβλεπτικού μας μοντέλου για την συνολική αξία πελάτη ενός και δύο μηνων.

Ακολουθώντας την ίδια τεχνική επιλογής βέλτιστου αριθμού με αυτή των κύριων συνιστωσών καταλήξαμε στην χρησιμοποίηση 20 singular values οι οποίες είχαν ως αποτέλεσμα την μείωση του RMSE από 0.779 σε 0.74 . Το ποσοστό μεταβλητότητας που εξηγείται από την χρησιμοποίηση 20 singular values ισούται με 30%. Τέλος πριν εφαρμοστεί η συγκεκριμένη μέθοδος κάθε στοιχείο του πίνακα Α διαιρέθηκε με το άθροισμα της αντίστοιχης γραμμής του ώστε το κάθε στοιχείο $i, j$ του πίνακα Α να υποδηλώνει την πιθανότητα αγοράς ενός προϊόντος $j$ από έναν πελάτη $i$.

## 5.5 Αποτελέσματα

Στη συγκεκριμένη ενότητα θα παρουσιαστούν τα αποτελέσματα των αλγορίθμων που εκπαιδεύτηκαν για την πρόβλεψη της απώλειας πελάτη ενός και δύο μπροστά και οι αλγόριθμοι που χρησιμοποιήθηκαν για την πρόβλεψη συνολικής αξίας πελάτη ενός και δύο μηνών. Στη συνέχεια θα γίνει μία ανάλυση στις μεταβλητές που συνέβαλαν περισσότερο στην πρόβλεψη των δύο αυτών προβλημάτων. Τέλος αφού επιλεγούν τα δύο καλύτερα μοντέλα για κάθε πρόβλημα θα σχηματιστούν προβλέψεις χρονικού ορίζοντα ενός και δύο μηνών.

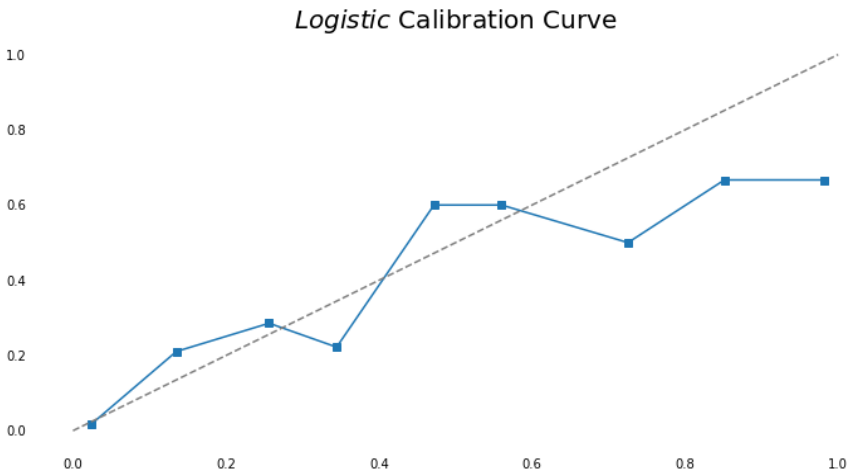
Αρχικά για τον ρυθμό απώλειας ενός μήνα οι αλγόριθμοι που εκπαιδεύσαμε ήταν μία λογιστική παλινδρόμηση και 3 αλγόριθμους δέντρων XGBoost LightGBM, RandomForest. Η σύγκριση τους έγινε χρησιμοποιώντας το AUC στο σύνολο επικύρωσης τα αποτελέσματα φαίνονται στον Πίνακα 5.14.

| Αλγόριθμος | AUC |
|---|---|
| XGBoost | 0.81 |
| LightGBM | 0.74 |
| Random Forest | 0.76 |
| Logistic Regression | 0.87 |

Πίνακας 5.14 Σύγκριση αλγορίθμων με το κριτήριο AUC

Από τα αποτελέσματα του Πίνακα 5.14 παρατηρούμε ότι η λογιστική παλινδρόμηση έχει το μεγαλύτερο AUC 0.87 και συνεπώς θα επιλεγεί για την δημιουργία προβλέψεων στην συνέχεια. Στη συνέχεια υπολογίζεται το AUC του συγκεκριμένου μοντέλου στο σύνολο ελέγχου. Το AUC βρέθηκε 0.84 το οποίο είναι αρκετά ικανοποιητικό.

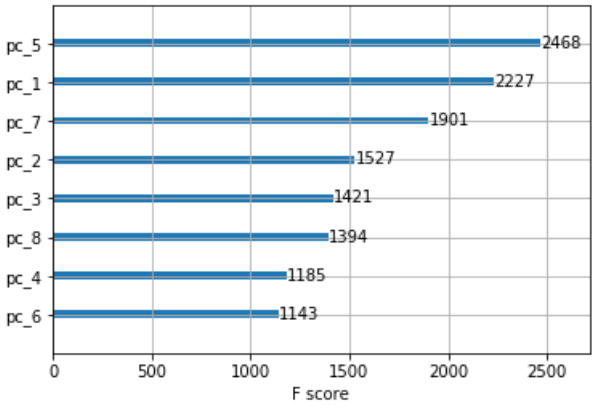
Στο συγκεκριμένο πρόβλημα δεν μας ενδιαφέρει μόνο η δημιουργία ενός αλγορίθμου που μπορεί να διαχωρίζει μεταξύ ομάδων. Ζωτικής σημασίας είναι και η ακρίβεια των πιθανοτήτων που εκτιμούντε από το μοντέλο καθώς οι ομάδες που έχουμε δεν είναι ντετερμινστικές αλλά στοχαστικές. Αυτό που μας ενδιαφέρει περισσότερο είναι η πιθανότητα ένας πελάτης να ανήκει σε κάποια από τις δύο ομάδες. Για παράδειγμα όταν προβλέπουμε ένας πελάτης έχει 50% πιθανότητα απώλειας γνωρίζουμε ότι έχοντας 100 πανομοιότυπους πελάτες σαν αυτόν οι 50 θα παραμείνουν πελάτες και οι 50 θα φύγουν. Ουσιαστικά θέλουμε ο αλγοριθμός μας για όσους πελάτες έχει προβλέψει πιθανότητα 50% να έχουν μείνει 50 πελάτες και να έχουν φύγει 50. Ένας τρόπος για να ελέξουμε την ακρίβεια των εκτιμημένων πιθανοτήτων είναι μέσω του Calibration Curve. Το συγκεκριμένο γράφημα δημιουργεί 10 κλάσεις από 0 εώς 0.1, από 0.1 εώς 0.2, κλπ μέχρι το 1 και χωρίζει τις πιθανότητες σε ανάλογες κλάσεις. Έπειτα υπολογίζει τον ρυθμό απώλειας πελατών σε κάθε κλάση και τα απεικονίζει σε ένα γράφημα ενώνοντας τα σημεία με μία γραμμή.




Σχήμα 5.7 Calibration Curve απώλειας ενός μήνα

Η διακεκομένη ευθεία είναι η $y = x$. Όταν τα σημεία μας βρίσκονται πάνω στην διακεκομένη ευθεία σημαίνει ότι οι εκτιμώμενος ρυθμός απώλειας ισούται με τον πραγματικό. Παρατηρούμε από το συγκεκριμένο γράφημα ότι οι πιθανότητες που εκτιμήσαμε μπορούν να χαρακτηρστούν ως ακριβείς.

Στη συνέχεια γίνεται αναφορά στις μεταβλητές που συντέλεσαν περισσότερο στην πρόβλεψη της απώλειας πελάτη. Οι μεταβλητές που χρησιμοποιήθηκαν ήταν 8 κύριες συνιστώσες όπως αυτές προέκυψαν από την ανάλυση κύριων συνιστωσών. Στο Σχήμα 5.8 φαίνεται η σημαντικότητα των μεταβλητών όπως προέκυψε από τον αλγόριθμο XGBoost.




Σχήμα 5.8 Σημαντικότητα μεταβλητών για την πρόβλεψη απώλειας πελάτη ενός μήνα

Από το Σχήμα 5.8 παρατηρούμε ότι η πέμπτη κύρια συνιστώσα βοηθάει περισσότερο στην πρόβλεψη της απώλειας πελάτη και στη συνέχεια η πρώτη. Έπεται η έβδομη κύρια συνιστώσα και ακολουθούν όλες οι υπόλοιπες. Για να δούμε ποιές μεταβλητές εμπεριέχονται στην πέμπτη,πρώτη και έβδομη συνιστώσα θα χρησιμοποιήσουμε τον συντελεστή συσχέτισης του Pearson μεταξύ των συνιστωσών και των αρχικών μεταβλητών που είχαμε.

| pc_5 | συσχέτιση |
|---|---|
| purchase_mean | 0.94 |
| last_purchase | 0.75 |

Πίνακας 5.15 Συντελεστής συσχέτισης μεταξύ πέμπτης κύριας συνιστώσας και αρχικών μεταβλητών.

Παρατηρούμε από τον Πίνακα 5.15 ότι η πέμπτη κύρια συνιστώσα που είναι και η πιο σημαντική ερμηνεύει κατά κύριο λόγο το χρονικό διάστημα που πέρασε από την τελευταία επίσκεψη (last_purchase) ενός πελάτη και την αλλιλεπίδραση της με τον μέσο όρο μεταξύ εμφανίσεων του (purchase_mean). Στην συνέχεια στον Πίνακα 5.16 φαίνεται ποίες μεταβλητές εμρηνεύει η πρώτη κύρια συνιστώσα.

| pc_1 | συσχέτιση |
|---|---|
| mean_btv | 0.89 |
| events | -0.83 |
| ma_5 | 0.83 |
| median_btv | 0.83 |
| events_lag2 | 0.80 |
| events_lag3 | 0.80 |
| events_lag1 | 0.79 |
| ma_3 | 0.79 |
| ewma_0.5 | 0.79 |
| std_btv | 0.78 |
| amount | -0.74 |

Πίνακας 5.16 Συντελεστής Συσχέτιστης μεταξύ πρώτης κύριας συνιστώσας και αρχικών μεταβλητών

Η πρώτη κύρια συνιστώσα ερμηνεύει κυρίως την αγοραστική συχνότητα των πελάτων, δηλαδή πόσες φορές επισκέπτηκαν κάποια κατάστημα, τον μέσο όρο μεταξύ επισκέψεων, πόσες επισκέψεις έκαναν τον τελευταίο προτελευταίο μήνα την τυπική απόκλιση των μεταξύ τους εμφανίσεων. Τέλος παρατηρούμε ότι ερμηνεύεται και η μεταβλητή συνολικό ποσό ανά πελάτη. Στον πίνακα 5.17 παρουσιάζονται οι μεταβλητές που εμηνεύονται από την έβδομη κύρια συνιστώσα.

| pc_7 | συσχέτιση |
|---|---|
| max_btv | 0.57 |
| std_btv | 0.44 |
| purchase_mean | 0.31 |
| events_lag1 | -0.28 |
| events_lag2 | -0.27 |
| events_lag3 | -0.27 |
| last_purchase | 0.26 |

Πίνακας 5.17 Συντελεστής Συσχέτιστης μεταξύ έβδομης κύριας συνιστώσας και αρχικών μεταβλητών.

Η έβδομη κύρια συνιστώσα ερμηνεύει πάλι μεταβλητές που αφορούν τις επισκέψεις πελατών σε καταστήματα όπως το μεγαλύτερο χρονικό διάστημα που πέρασε μεταξύ δύο επισκέψεων, η τυπική απόκλιση και η αλλιλεπίδραση χρονου που έχει περάσει από την τελευταία του επίσκεψη. Επίσης ερμηνεύεται και ο αριθμός των συνολικών επισκέψεων ενός, δύο και τρίων μηνών στο παρελθόν.

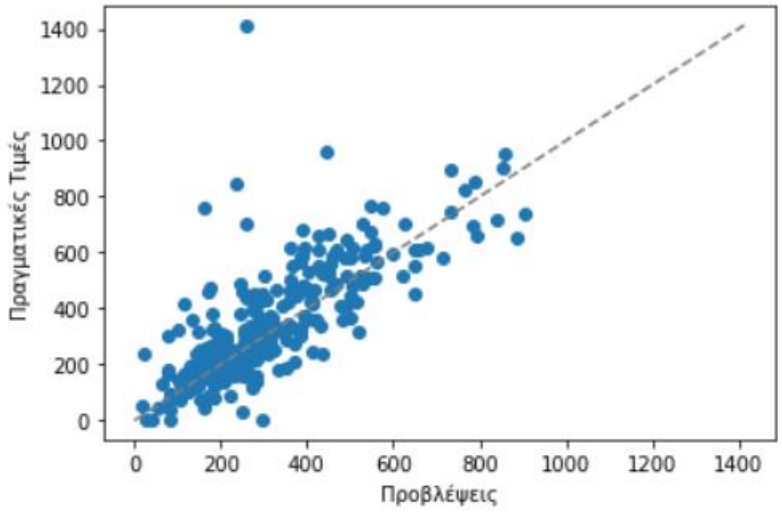
Από την ερμηνεία των κυρίων συνιστωσών που βοήθησαν στην πρόβλεψη απώλειας πελάτη παρατηρούμε ότι οι σημαντικότερες μεταβλητές είναι εκείνες που απεικονίζουν την αγοραστική συχνότητα ενός πελάτη, το χρονικό διάστημα που περνά μεταξύ των επισκέψεων του και πότε πραγματοποίησε την τελευταία του εμφάνιση. Μεταβλητές που χαρακτηρίζουν το καλάθι δεν έχουν ιδαίτερη συμβολή στην πρόβλεψη της απώλειας πελάτη.

Στη συνέχεια γίνεται αναφορά στους αλγόριθμους που χρησιμοποιήθηκαν για την πρόβλεψη της συνολικής αξίας πελάτη και γίνεται επιλογή του καλύτερου βάση του κριτηρίου RMSE. Οι αλγόριθμοι που εκπαιδεύσαμε ήταν οι RandomForest, XGBoost, LightGBM και Γραμμική Παλιδρόμηση. Στον πίνακα 5.18 παρουσιάζονται τα αποτελέσματα του RMSE για κάθε αλγόριθμο.

| Αλγόριθμος | RMSE |
|---|---|
| XGBoost | 0.86 |
| LightGBM | 0.81 |
| RandomForest | 0.87 |
| Γραμμική Παλινδρόμηση | 1.23 |

Πίνακας 5.18 Σύγκριση αλγορίθμων με το κριτήριο RMSE

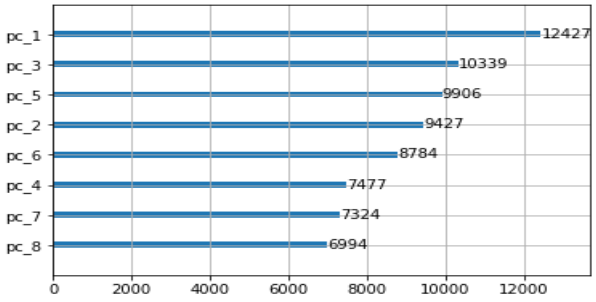
Ο αλγόριθμος LightGBM απέφερε τα καλύτερα αποτελέσματα στο σύνολο επικύρωσης οπότε θα τον χρησιμοποιήσουμε για την πρόβλεψη της συνολικής αξίας πελάτη ενός μήνα. Στο Σχήμα 5.9 φαίνεται και το διάγραμμα διασποράς των πραγματικών και εκτιμώμενων τιμών στο σύνολο ελέγχου.




Σχήμα 5.9 Διάγραμμα διασποράς πραγματικών με εκτιμώμενων τιμών

Από το διάγραμμα διασποράς παρατηρούμε ότι ο αλγόριθμος μας προβλέπει ικανοποιητικά την συνολική αξία πελάτη σε δεδομένα που δεν έχει εκπαιδευτεί. Η διακεκομένη γραμμή είναι η $y = x$ σημεία που πέφτουν πάνω στην ευθεία σημαίνει ότι ο αλγόριθμος μας τα προέβλεψε

ακριβώς. Έπειτα στον Σχήμα 5.10 παρουσιάζονται οι κύριες συνιστώσες που βοήθησαν στην πρόβλεψη της συνολικής αξίας πελάτη.




Σχήμα 5.10 Σημαντικότητα μεταβλητών για την πρόβλεψη συνολικής αξίας πελάτη ενός μήνα

Παρατηρούμε ότι οι πρώτη, τρίτη, πέμπτη και δεύτερη κύριες συνιστώσες βοηθούν στην πρόβλεψη της CLV. Για να δούμε ποιές μεταβλητές εμπεριέχουν θα χρησιμοποιήσουμε πάλι τον συντελεστή συσχέτισης του Pearson όπως στην πρόβλεψη απώλειας πελάτη. Στον Πίνακα 5.19 παρουσιάζονται οι συσχετίσεις των αρχικών μεταβλητών με την πρώτη κύρια συνιστώσα.

| pc_1 | συσχέτιση |
|---|---|
| mean_btv | -0.89 |
| events | 0.83 |
| ma_5 | -0.83 |
| median_btv | -0.83 |
| events_lag2 | 0.80 |
| events_lag3 | 0.80 |
| events_lag1 | 0.79 |
| ma_3 | -0.79 |
| ewma_0.5 | -0.79 |
| std_btv | -0.78 |
| amount | -0.74 |

Πίνακας 5.19 Συντελεστής Συσχέτιστης μεταξύ πρώτης κύριας συνιστώσας και αρχικών μεταβλητών

Η πρώτη κύρια συνιστώσα ερμηνέυει κυρίως μεταβλητές που έχουν να κάνουν με την συχνότητα επίσκεψης και τον μέσο όρο μεταξύ επισκέψεων. Στην συνέχεια στον Πίνακα 5.20 παρουσιάζονται οι συσχετίσεις των αρχικών μεταβλητών με την τρίτη κύρια συνιστώσα.

| pc_3 | συσχέτιση |
|---|---|
| last_purchase | 0.75 |
| purchase_mean | 0.75 |
| probs | 0.75 |
| active_days | -0.51 |

Πίνακας 5.20 Συντελεστής συσχέτιστης μεταξύ τρίτης κύριας συνιστώσας και αρχικών μεταβλητών

Η τρίτη κύρια συνιστώσα εμπεριέχει μεταβλητές που έχουν σχέση με το χρονικό διάστημα που πέρασε από την τελευταία εμφάνιση ενός πελάτη και την αλλιλεπίδρασή του με την μέση του τιμή μεταξύ εμφανίσεων, την πιθανότητα απώλειας και το συνολικό αριθμό ημερών που ένας πελάτης είναι ενεργός. Στον Πίνακα 5.21 φαίνονται οι συχετίσεις των αρχικών μεταβλητών με την πέμπτη κύρια συνιστώσα.

| pc_5 | συσχέτιση |
|---|---|
| mode_basket | 0.77 |
| min_basket | 0.77 |
| active_days | 0.45 |

Πίνακας 5.21 Συντελεστής συσχέτιστης μεταξύ πέμπτης κύριας συνιστώσας και αρχικών μεταβλητών

Περιλαμβάνει μεταβλητές όπως συνηθέστερη αξία καλαθιού μικρότερο καλάθι και συνολικό αριθμό ημερών που ένας πελάτης είναι ενεργός. Τέλος δίνονται στον Πίνακα 5.22 οι συχετίσεις των αρχικών μεταβλητών με την δεύτερη κύρια συνιστώσα.

| pc_2 | συσχέτιση |
|---|---|
| mean_basket | 0.84 |
| std_basket | 0.80 |
| median_basket | 0.79 |
| ma_basket3 | 0.76 |
| ewma_basket0.5 | 0.75 |
| max_basket | 0.73 |
| amount | 0.70 |
| amount_lag3 | 0.61 |
| amount_lag1 | 0.59 |
| amount_lag2 | 0.58 |

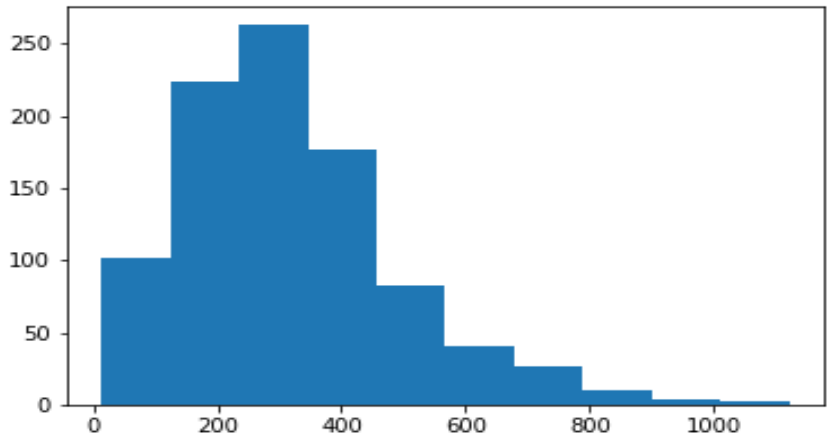
Πίνακας 5.22 Συντελεστής συσχέτιστης μεταξύ δεύτερης κύριας συνιστώσας και αρχικών μεταβλητών

Η δεύτερη κύρια συνιστώσα ερμηνεύει μεταβλητές που αφορούν το καλάθι και το συνολικό ποσό. Στη συνέχεια παρουσιάζονται οι προβλέψεις που διαμορφώθηκαν για την απώλεια και συνολική αξία πελάτη ενός μήνα. Αρχικά παρουσιάζονται οι προβλέψης για την απώλεια πελάτη στον Πίνακα 5.23 κατά φθίνουσα σειρά.

| mean_btv | last_purchase | probability |
|---|---|---|
| 1.2 | 113 | 1 |
| 2.1 | 111 | 0.99 |
| 1.5 | 68 | 0.99 |
| 5.4 | 92 | 0.99 |
| 18.84 | 91 | 0.99 |
| 1.3 | 60 | 0.99 |
| 20.69 | 61 | 0.99 |
| 8.7 | 71 | 0.95 |
| 11 | 35 | 0.94 |
| 3.6 | 61 | 0.92 |
| 33.5 | 0 | 0.92 |
| 14.6 | 16 | 0.73 |
| 3.8 | 48 | 0.72 |
| 2.4 | 42 | 0.69 |

Πίνακας 5.23 Προβλέψεις απώλειας πελάτη ενός μήνα

Στο Σχήμα 5.11 παρουσιάζεται η κατανομή της προβλεπόμενης CLV.



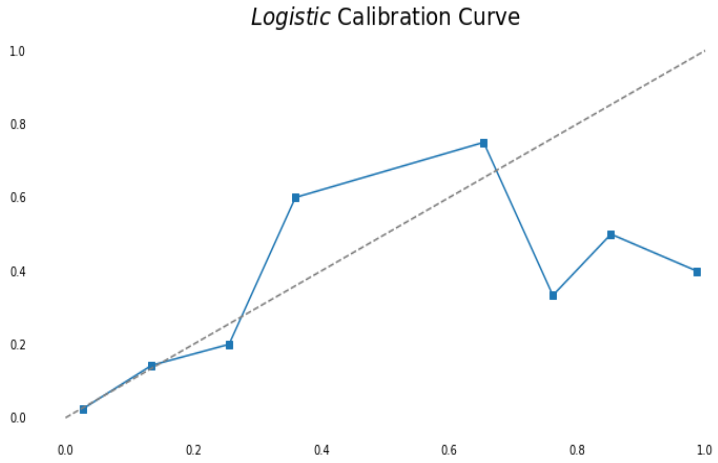

Σχήμα 5.11 Κατανομή CLV για τον μήνα Μάρτιος 2017

Στον Πίνακα 5.24 δίνονται και κάποια βασικά περιγραφικά στατιστικά της προβλεπόμενης CLV.

| | |
|---|---|
| count | 930 |
| mean | 317.14 |
| std | 177.83 |
| min | 12.79 |
| 1Q | 198.31 |
| median | 287.79 |
| 3Q | 410.16 |
| max | 1121.74 |

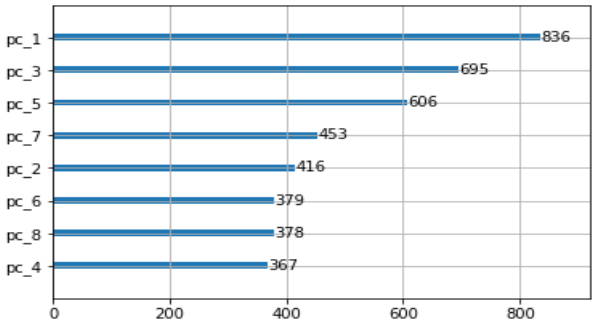
Πίνακας 5.24 Περιγραφικά στατιστικά της προβλεπόμενης CLV

Στη συνέχεια εκπαιδεύουμε μία λογιστική παλινδρόμιση για την πρόβλεψη απώλειας πελάτη δύο μηνών. Το AUC στο σύνολο ελέγχου ισούται με 0.73 σημαντικά μικρότερο από εκείνο του ενός μήνα. Αναμέναμε τη μείωση στην απόδοση του μοντέλου μας στους δύο μήνες πρόβλεψης καθώς για την πρόβλεψη του churn ενός μήνα ιδιαίτερα σημαντικές μεταβλητές ήταν όσες εξηγούσαν την συμπεριφορά του πελάτη στις τελευταίες του επισκέψεις. Στο Σχήμα 5.12 παρουσιάζεται και το Calibration Curve.




Σχήμα 5.12 Calibration Curve ρυθμού απώλειας δύο μηνών

Στο Σχήμα 5.13 δίνονται και η σημαντικότητα των μεταβλητών.



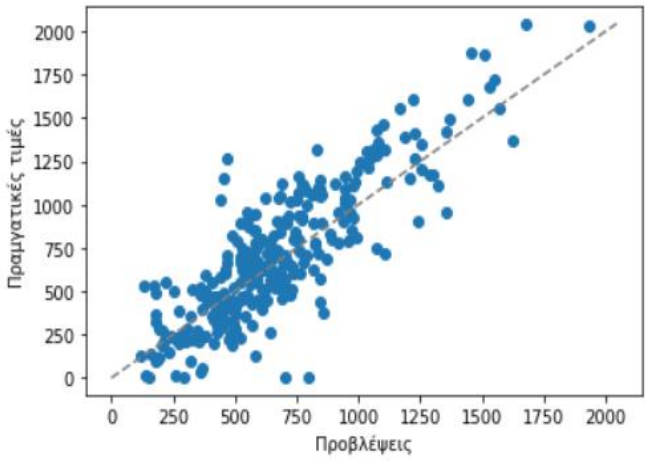

Σχήμα 5.13 Σημαντικότητα μεταβλητών ρυθμού απώλειας δύο μηνών.

Από το Σχήμα 5.13 παρατηρούμε ότι οι ίδιες συνιστώσες επηρεάζουν την μεταβλητή απόκρισης για ένα και δύο μήνες με ελάχιστες διαφορές. Η πρώτη κύρια συνιστώσα είναι πιο ημαντική αντί για την πέμπτη. Όπως είδαμε πιο πριν η πρώτη κύρια συνιστώσα ερμηνεύει μεταβλητές που αφορούν την αγοραστική συχνότητα ενός πελάτη και η τρίτη πόσο χρονικό διάστημα πέρασε από την τελευταία επίσκεψη και την αλλιλεπίδραση του με τον μέσο όρο μεταξύ επισκέψεων. Δίνονται και οι προβλέψεις του μοντέλου στον Πίνακα 5.25 κατά αύξουσα σειρά.

| mean_btv | last_purchase | probability |
|---|---|---|
| 1.2 | 113 | 1 |
| 2.1 | 111 | 0.99 |
| 1.3 | 60 | 0.99 |
| 1.6 | 68 | 0.99 |
| 19.63 | 91 | 0.99 |
| 5.43 | 92 | 0.97 |
| 33.5 | 0 | 0.93 |
| 17.17 | 16 | 0.93 |
| 11.08 | 35 | 0.90 |
| 12.44 | 91 | 0.85 |

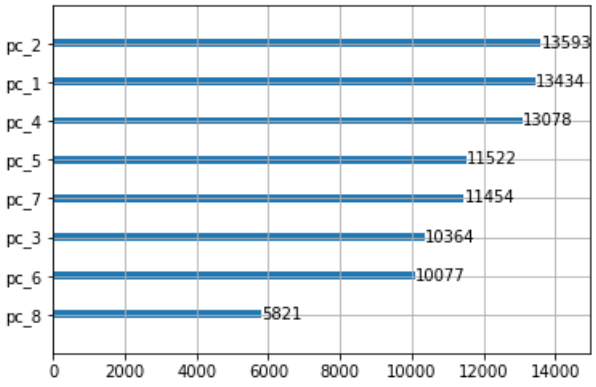
Πίνακας 5.25 Προβλέψεις απώλειας πελάτη δύο μηνών.

Για την συνολική αξία πελάτη δύο μηνών εκπαιδεύουμε έναν αλγόριθμο LightGBM. Το παράδοξο είναι ότι για την πρόβλεψη της CLV δύο μηνών ο αλγόριθμος αποφέρει πολύ καλύτερα αποτελέσματα από ότι για ένα μήνα. Στο Σχήμα 5.14 δίνεται το διάγραμμα διασποράς μεταξύ των εκτιμόμενων προβλέψεων της CLV και των πραγματικών.




Σχήμα 5.14 Διάγραμμα διασποράς πραγματικών με εκτιμώμενων τιμών

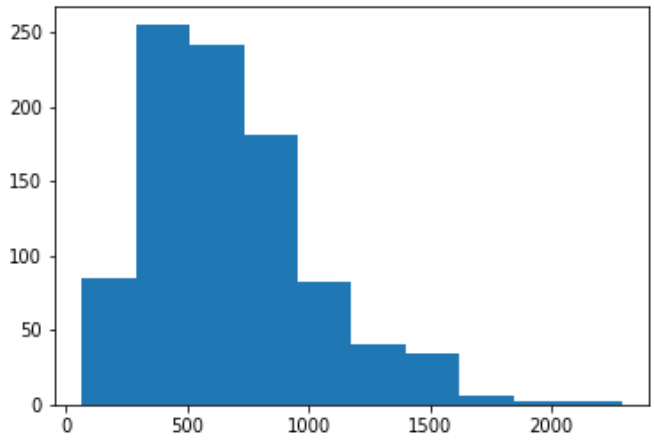
Στο Σχήμα 5.15 απεικονίζεται η σημαντικότητα των μεταβλητών για την πρόβλεψη της CLV.




Σχήμα 5.15 Σημαντικότητα μεταβλητών για την πρόβλεψη της CLV δύο μηνών.

Γιά την πρόβλεψη της CLV δύο μηνών παρατηρούμε ότι η δεύτερη κύρια συνιστώσα, που όπως είδαμε πριν εμπεριέχει πληροφορία για τα καλάθια του χρήστη και πόσα συνολικά έχει

ξοδέψει, είναι η πιο σημαντική μεταβλητή. Η πρώτη κύρια συνιστώσα είναι είναι η δεύτερη πιο σημαντική μεταβλητή που ερμηνεύει την αγοραστική συχνότητα ενός πελάτη (εμφανίσεις, μέσος μεταξύ εμφανίσεων, κλπ). Στο Σχήμα 5.16 παρουσιάζεται και η κατανομή της προβλεπόμενης CLV δύο μηνων.




Σχήμα 5.16 Κατανομή CLV για τους μήνες Μάρτιος,Απρίλιος 2017

Συγκρίνοντας την με την κατανομή της CLV για τους μήνες Ιανουάριος, Φεβρουάριος (βλ. Σχήμα 5.5) παρατηρούμε ότι οι δύο κατανομές είναι σχεδόν όμοιες. Δίνονται στον Πίνακα 5.26 βασικά περιγραφικά στατιστικά της προβλεπόμενης CLV δύο μηνών.

| | |
|---|---|
| count | 930 |
| mean | 678.36 |
| std | 342.88 |
| min | 66.95 |
| 1Q | 443.71 |
| median | 613.96 |
| 3Q | 862.46 |
| max | 2287.47 |

Πίνακας 5.26 Περιγραφικά στατιστικά της προβλεπόμενης CLV δύο μηνών.

Από τον Πίνακα 5.26 παρατηρούμε ότι το μοντέλο μας προβλέπει σαν μικρότερη τιμή το 66.95. Γεγονός που συμβαίνει καθώς το σύνολο δεδομένων μας έχει ελάχιστους πελάτες με μηδενική CLV τους μήνες που χρησιμοποιήθηκαν για την εκπαίδευση του μοντέλου μας (Ιανουάριος, Φεβρουάριος).

# 6. Συμπεράσματα και συζήτηση

Στόχος της παρούσας διπλωματικής είναι κατά κύριο λόγο η πρόβλεψης της συνολικής αξίας και απώλειας πελάτη. Οι τεχνικές που χρησιμοποιήθηκαν βασίζονται σε δέντρα παλινδρόμησης και κατηγοριοποίησης με τις σύγχρονες επεκτάσεις τους. Επίσης έγινε αναφορά στην σημαντικότητα της εξαγωγής και σωστής επιλογής μεταβλητών με σκοπό να βοηθήσουν στην πρόβλεψη της συνολικής αξίας και απώλειας πελάτη. Έχοντας στην κατοχή του ο αναλυτής μεταβλητές ισχυρά συσχετισμένες με την μεταβλητή απόκρισης συνήθως οδηγούν σε ικανοποιητική προσαρμογή στις τιμές της μεταβλητής απόκρισης ανεξαρτήτως μοντέλου.

Η συνολική αξία πελάτη είναι ένα μη γραμμικό πρόβλημα και όπως είδαμε σε προηγούμενη ενότητα η επίλυση του με γραμμικά υποδείγματα απέτυχε πλήρως. Ο αλγόριθμος LightGBM αποτελεί μια σύγχρονη επέκταση των τεχνικών βασισμένες σε δέντρα, ο οποίος χρησιμοποιείται ευρέως. Στην πρόβλεψη της συνολική αξίας πελάτη ο συγκεκριμένος αλγόριθμος πέτυχε τα καλύτερα αποτελέσματα. Παρουσίασε καλύτερη προσαρμογή στην πρόβλεψη της CLV από ότι ο αλγόριθμος Random Forest που συνήθως χρησιμοποείται για το συγκεκριμένο πρόβλημα. ([4],[10]). Κάτι που έχει ιδιαίτερο ενδιαφέρον να αναφέρουμε, είναι πως όταν χρησιμοποιήθηκε για πρόβλεψη δύο μηνών μπροστάα απέφερε καλύτερα αποτελέσματα. Η εποχικότητα που παρουσιάζουν οι χρονοσειρές μας λογικά έπαιξε καθοριστικό ρόλο στην καλύτερη προσαρμογή των τιμών της CLV δύο μηνών μπροστά.

Η πρόβλεψη της απώλειας πελάτη είναι ένα ιδαιέτερα δύσκολο πρόβλημα. Η πρώτη δυσκολία που έχει να αντιμετωπίσει κανείς είναι ο ορισμός της απώλειας πελάτη καθώς δεν υπάρχει κάποιος κανόνας που να την ορίζει σε μη συνδρομιτικές επιχειρήσεις όπως στην δική μας περίπτωση. Δοκιμάστηκαν διάφορες προσεγγίσεις με αυτή των 14 ημερών (πελάτης που πέρασε χρονικό διάστημα μεγαλύτερο των 14 ημερών από την τελευταία του εμφάνιση ορίζεται σαν απώλεια) να αποφέρει τα καλύτερα αποτελέσματα. Η λογιστική παλινδρόμηση παρουσίασε καλύτερη προσαρμογή στην πρόβλεψη απώλειας πελάτη. Όχι μόνο στον διαχωρισμό των δύο κατηγοριών αλλά και στην ακρίβεια των εκτιμημένων πιθανοτήτων απώλειας πελάτη. Τέλος σημαντική διαφορά είχε η απόδοση του μοντέλου μας όταν χρησιμοποιήθηκε για την πρόβλεψη απώλειας πελάτη δύο μηνών, καθώς όπως είδαμε σε

προηγούμενη ενότητα μεταβλητές που ερμηνεύουν την συμπεριφορά πελάτων πρόσφατα και όχι καθόλη τη διάρκεια που αλλιλεπιδρούν με την επιχείρηση είναι ιδαιέτερα σημαντικές.

Κλείνοντας, γίνεται αναφορά σε τεχνικές που θα είχε ιδαιέτερο ενδιαφέρον η εφαρμογή τους εάν είχαμε στην κατοχή μας περισσότερες παρατηρήσεις. Όσον αφορά στην απώλεια πελάτη θα μπορούσε να χρησιμοποιηθεί μία διαφορετική συνάρτηση ζημιάς από ότι το cross entropy. Σε προβλήματα κατηγοριοποίησης δίτιμων μεταβλητών απόκρισης με άνισες κλάσεις για την αντιμετώπιση του συγκεκριμένου προβλήματος έχουν προταθεί πληθώρα συναρτήσεων ζημιάς που προσπαθούν να προσεγγίσουν το AUC. Το κριτήριο AUC δεν είναι παραγωγίσιμο και σε προηγούμενη ενότητα είδαμε ότι οι συναρτήσεις ζημιάς στους αλγορίθμους LightGBM και XGBoost αλλά και στην λογιστική παλινδρόμηση οφείλουν να είναι παραγωγίσημες. Έχουν προταθεί διάφορες συναρτήσεις ζημιάς που προσεγγίζουν το AUC ([6],[15]). Το αρνητικό με τις συγκεκριμένες τεχνικές είναι ότι ενώ συντελούν στην αύξηση του AUC, δεν δίνουν ικανοποιητικά αποτελέσματα στην ακρίβεια των εκτιμημένων πιθανοτήτων. Υπάρχουν τεχνικές που διορθώνουν το συγκεκριμένο πρόβλημα αλλά απαιτούν δείγμα το λιγότερο 1000 παρατηρήσεων το οποίο δεν έχει χρησιμοποιηθεί για την εκπαίδευση του αλγορίθμου. Μία τέτοια τεχνική είναι η Isotonic Regression ([7]). Επίσης για την εξαγωγή μεταβλητών από την αλλιλεπίδραση πελατών με προϊόντα χρησιμοποιούνται αλγόριθμοι που είναι ιδαιέτερα διαδεδομένοι στην ερμηνεία φυσικής γλώσσας πεδίο στο οποίο έχει πραγματοποιηθεί ένας μεγάλος αριθμός δημοσιεύσεων ([10],[21]). Τέλος θα είχε ιδαίτερο ενδιαφέρον η προσπάθεια επίλυσης του προβλήματος με σύγχρονες τεχνικές ανάλυσης χρονολογικών σειρών.

# Παράρτημα

## Κώδικες

Η γλώσσα προγραματισμού Python χρησιμοποιήθηκε για την δημιουργία των σχημάτων και αποτελεσμάτων που παρουσιάζονται στην ενότητα 5

**Πίνακας 5.1**

```
# Checking which day of the week has the most visits

day_freq=collections.Counter(data['dayofWeek'])

day_freq.most_common()
```

**Σχήμα 5.1**

```
sales.plot(x='Date',y='total_sales')

plt.ylim(280000,350000)

plt.title('Συνολικές Πωλήσεις ανά μήνα');´
```

**Πίνακας 5.2**

```
customer_df.amount.describe()
```

**Σχήμα 5.2**

```
# plotting the total amount spend of each customer

plt.hist(customer_df['amount'],bins=15);
```

**Πίνακας 5.3**

```
customer_df['mean_btv'].describe()
```

**Πίνακας 5.4**

```
customer_df['events'].describe()
```

**Πίνακας 5.5**

```
customer_df['last_purchase'].describe()
```

**Σχήμα 5.3**

```
customer_df['churn'].value_counts().plot(kind='bar');
```

**Σχήμα 5.4**

```
plt.hist(customer_df['CLV_1'].fillna(0));
```

**Πίνακας 5.6**

```
customer_df['CLV_1].fillna(0).describe()
```

**Σχήμα 5.5**

```
plt.hist(customer_df['CLV_2'].fillna(0));
```

**Πίνακας 5.7**

```
customer_df['CLV_2].fillna(0).describe()
```

**Προπαρασκευή δεδομένων**

```
days=my_data.sort_values(by=['Transaction_date']).drop_duplicates('Basket_id')

my_data['previous_visit'] = days.groupby('Customer_no').Transaction_date.shift()
```

```
my_data['days_btw_visits'] =my_data['Transaction_date']-my_data['previous_visit']
 my_data['days_btw_visits'] = my_data['days_btw_visits'].apply(lambda x: x.days)
def ma_calc(df,period):
    calc_df = df.groupby('Customer_no').apply(lambda x:
x['days_btw_visits'].rolling(period).mean()).reset_index().shift().dropna()
    calc_df = calc_df.groupby('Customer_no').tail(1)
    calc_df = calc_df.rename(columns={
        'days_btw_visits':'ma_'+str(period)
        })
    return calc_df.drop('level_1',axis=1)
def ewma_calc(df,alpha):
    calc_df = df.groupby('Customer_no').apply(
        lambda x: x['days_btw_visits'].ewm(
            alpha=alpha).mean().tail(1)).reset_index()
    calc_df = calc_df.rename(columns={
        'days_btw_visits':'ewma_'+str(alpha)
        })
    return calc_df.drop('level_1',axis=1)
def ewma_calc_basket(df,alpha):
    calc_df = df.groupby('Customer_no').apply(
        lambda x: x['item_net_amount'].ewm(
            alpha=alpha).mean().tail(1)).reset_index()
    calc_df = calc_df.rename(columns={
        'item_net_amount':'ewma_basket'+str(alpha)
```

```
        })
    return calc_df.drop('level_1',axis=1)
def ma_calc_basket(df,period):
    calc_df = df.groupby('Customer_no').apply(lambda x:
x['item_net_amount'].rolling(period).mean()).reset_index().shift().dropna()
    calc_df = calc_df.groupby('Customer_no').tail(1)
    calc_df = calc_df.rename(columns={
        'item_net_amount':'ma_basket'+str(period)
        })
    return calc_df.drop('level_1',axis=1)
def month_amount_lagged(data,lag):


    last_amount = data.groupby(
        ['Customer_no',
'Basket_id','Transaction_date_month']).item_net_amount.sum().reset_index()
    last_amount =
last_amount.groupby(['Customer_no','Transaction_date_month']).item_net_amount.sum().res
et_index()
    filt = last_amount['Transaction_date_month'].max() - relativedelta(
            months=lag-1) == last_amount['Transaction_date_month']
    last_amount = last_amount[filt]
    last_amount = last_amount.rename(columns={
        'item_net_amount': 'amount_lag'+str(lag)
    })
    return last_amount.drop('Transaction_date_month',axis=1)
```

```
def month_events_lagged(data,lag):


   last_events =
data.groupby(['Customer_no','Transaction_date_month']).Basket_id.nunique().reset_index()
   filt = last_events['Transaction_date_month'].max() - relativedelta(
         months=lag-1) == last_events['Transaction_date_month']
   last_events = last_events[filt]
   last_events = last_events.rename(columns={
      'Basket_id':'events_lag'+str(lag)
   })
   return last_events.drop('Transaction_date_month',axis=1)
def labels_calc(df,date):
   filter_day = df['Transaction_date'].max() - relativedelta(
         months=date)
   filter_labels = df['Transaction_date'] >= filter_day
      # Obtaining the labels of CLV
   CLV = df[filter_labels].groupby(
      'Customer_no').item_net_amount.sum().reset_index()
   CLV = CLV.rename(columns={'item_net_amount': 'CLV'})


   # Obtaining the labels of CLV
   last_p = df['Transaction_date'].max() - df.groupby(
      'Customer_no').Transaction_date.max()
   last_p = last_p.apply(lambda x: x.days)
```

```
    churn = (last_p > 14).astype(int).reset_index()
    churn = churn.rename(columns={'Transaction_date': 'churn'})


    label_df = CLV.merge(churn, on='Customer_no',how='right')
    label_df = label_df.fillna(0)
    return label_df
def df_merge(a,b,c,d):
    df = a.merge(b, on='Customer_no')
    df = df.merge(c, on='Customer_no')
    df = df.merge(d, on='Customer_no')
    return df
def df_merge_predict(a,b,c):
    df = a.merge(b, on='Customer_no')
    df = df.merge(c, on='Customer_no')
    return df
def df_slice(df,date):
    filter_day = df['Transaction_date'].max() - relativedelta(
            months=date)
    filter_train = df['Transaction_date'] < filter_day
    df = df[filter_train]
    return df
def df_slice_predict(df,date):
    filter_day = df['Transaction_date'].min() + relativedelta(
```

```
            months=date)
    filter_predict = df['Transaction_date'] >= filter_day
    df = df[filter_predict]
    return df
def preprocessing_visits(data):
    # first last order and number of active days
    end_time = np.datetime64(data['Transaction_date'].max())
    begin_time = np.datetime64(data['Transaction_date'].min())
    customer_df = data.groupby(
        'Customer_no', as_index=False).Transaction_date.agg(
            {np.max, np.min, np.ptp}).reset_index()
    customer_df['last_purchase'] = end_time - customer_df['amax']
    customer_df['first_purchase'] = customer_df['amin'] - begin_time
    customer_df['ptp'] = customer_df['ptp'].apply(lambda x: x.days)
    customer_df['last_purchase'] = customer_df['last_purchase'].apply(
        lambda x: x.days)
    customer_df['first_purchase'] = customer_df['first_purchase'].apply(
        lambda x: x.days)
    customer_df = customer_df.rename(columns={'ptp': 'active_days'})
    customer_df = customer_df.drop(['amax', 'amin'], axis=1)
   # Visits of each customer
    frequency_df = data.groupby(
        'Customer_no').Basket_id.nunique().reset_index()
```

```
frequency_df = frequency_df.rename(columns={'Basket_id': 'events'})
customer_df = customer_df.merge(frequency_df, on='Customer_no')


# Statistics of days between visits
kur = data[['days_btw_visits', 'Customer_no']].dropna()
stats_of_btw_days = kur.groupby('Customer_no').days_btw_visits.agg([
    np.mean, np.median, np.std, np.min, np.max, skew, kurtosis,
    lambda x: mode(x)[0][0], count_zeros
])
stats_of_btw_days = stats_of_btw_days.rename(
    columns={
        'mean': 'mean_btv',
        'median': 'median_btv',
        'std': 'std_btv',
        'amin': 'min_btv',
        'amax': 'max_btv',
        'skew': 'skew_btv',
        'kurtosis': 'kur_btv',
        '<lambda>': 'mode_btv'
    })
# Calculating Ewmas with alpha=0.9, 0.5
ewma_09 = ewma_calc(kur,0.9)
ewma_05 = ewma_calc(kur,0.5)
```

```
    customer_df = customer_df.merge(ewma_09, on='Customer_no')
    customer_df = customer_df.merge(ewma_05, on='Customer_no')


    # Calculating MA 3 and 5 periods
    ma_3  =ma_calc(kur,3)
    ma_5  =ma_calc(kur,5)
    customer_df = customer_df.merge(ma_3, on='Customer_no')
    customer_df = customer_df.merge(ma_5, on='Customer_no')
    customer_df = customer_df.merge(stats_of_btw_days, on='Customer_no')
   # customers last month's visits lagged
    f=[1,2,3]
    for i in f:
        last_ev =month_events_lagged(data,i)
        customer_df = customer_df.merge(last_ev, on='Customer_no', how='left')
    # last purchase mean btv interaction


    customer_df['purchase_mean']=customer_df['last_purchase']/customer_df['mean_btv']


    customer_df = customer_df.fillna(0)
    return customer_df
def preprocessing_basket(data):
  #Total amount spend,mean of baskets,std of baskets,min,max and skewness basket
   basket_df = data.groupby(
```

```
    ['Customer_no', 'Basket_id']).item_net_amount.sum().reset_index()
amount_df = basket_df.groupby('Customer_no').item_net_amount.agg([
    np.mean, np.median, np.std, np.min, np.max, 'sum', skew, kurtosis,
    lambda x: mode(x)[0][0]
])
amount_df = amount_df.rename(
    columns={
        'sum': 'amount',
        'mean': 'mean_basket',
        'median': 'median_basket',
        'std': 'std_basket',
        'amin': 'min_basket',
        'amax': 'max_basket',
        'skew': 'skew_basket',
        'kurtosis': 'kur_basket',
        '<lambda>': 'mode_basket'
    })

# Finding how many different products bought by customers
products_bought = data.groupby('Customer_no').EAN.nunique().reset_index()
products_bought = products_bought.rename(columns={'EAN': 'diff_products'})

amount_df = amount_df.merge(products_bought, on='Customer_no')
```

```
  #Time series features


     # Calculating Ewmas with alpha=0.9, 0.5
   ewma_09 = ewma_calc_basket(basket_df,0.9)
  ewma_05 = ewma_calc_basket(basket_df,0.5)
  amount_df = amount_df.merge(ewma_09, on='Customer_no')
  amount_df = amount_df.merge(ewma_05, on='Customer_no')


     # Calculating MA 3 and 5 periods
  ma_3  =ma_calc_basket(basket_df,3)
  ma_5  =ma_calc_basket(basket_df,5)
  amount_df = amount_df.merge(ma_3, on='Customer_no')
  amount_df = amount_df.merge(ma_5, on='Customer_no')
  f=[1,2,3]
  for i in f:
     last_am =month_amount_lagged(data,i)
     amount_df = amount_df.merge(last_am, on='Customer_no', how='left')
     amount_df = amount_df.fillna(0)
 return amount_df
def entropy_calc(data):
# Store entropy
  store_entropy = data.groupby(['Customer_no','Store_number']).Basket_id.nunique()
```

```
    store_entropy = store_entropy.groupby('Customer_no').apply(lambda
x:entropy(x/sum(x))).reset_index()
    store_entropy = store_entropy.rename(columns={'Basket_id':'store_entr'})
    # Month entropy
    month_entropy =
data.groupby(['Customer_no','Transaction_date_month']).Basket_id.nunique()
    month_entropy = month_entropy.groupby('Customer_no').apply(lambda
x:entropy(x/sum(x))).reset_index()
    month_entropy = month_entropy.rename(columns={'Basket_id':'month_entr'})

    # Day entropy

    day_entropy = data.groupby(['Customer_no','dayofWeek']).Basket_id.nunique()
    day_entropy = day_entropy.groupby('Customer_no').apply(lambda
x:entropy(x/sum(x))).reset_index()
    day_entropy = day_entropy.rename(columns={'Basket_id':'day_entr'})

    entropy_df = store_entropy.merge(month_entropy,on='Customer_no')
    entropy_df = entropy_df.merge(day_entropy,on='Customer_no')

    return entropy_df
```

**Πίνακας 5.8**

```
cor = np.abs(customers_train.drop('Customer_no',axis=1).corr())
cor['churn'].sort_values(ascending=False)
```

**Random Forest σημαντικότητα**

```
def random_forest_importances(df,target):
    X = df.loc[:, df.columns != target]
    X = X.drop('Customer_no', axis = 1)
    y = df[target]
    clf = RandomForestClassifier(n_estimators=200)
    clf.fit(X,y)


    important_features = pd.Series(data=clf.feature_importances_,index=X.columns)
    print(important_features.sort_values(ascending=False))
```

**Πίνακας 5.9**

```
# Creating a Random Forest to see the feature importances
random_forest_importances(customers_train.drop('CLV',axis=1),'churn')
```

**Συσχέτιση μεταξύ επεξηγηματικών μεταβλητών**

```
def get_redundant_pairs(df):
    '''Get diagonal and lower triangular pairs of correlation matrix'''
    pairs_to_drop = set()
    cols = df.columns
    for i in range(0, df.shape[1]):
        for j in range(0, i+1):
```

```
            pairs_to_drop.add((cols[i], cols[j]))
    return pairs_to_drop


def get_top_abs_correlations(df, n=5):
    au_corr = df.corr().abs().unstack()
    labels_to_drop = get_redundant_pairs(df)
    au_corr = au_corr.drop(labels=labels_to_drop).sort_values(ascending=False)
    return au_corr[0:n]
```

**Πίνακας 5.10**

```
get_top_abs_correlations(customers_train.drop('Customer_no',axis=1),20)
```

**Δημιουργία συνόλου εκπαίδευσης,επικύρωσης και ελέγχου**

```
def train_valid_test_split(df,perc,target):
    X = df.loc[:, df.columns != target]
    y = df[target]
    X_train, X_test, y_train, y_test = train_test_split(X, y, stratify=y,test_size=perc)
    X_train, X_val, y_train, y_val  = train_test_split(X_train, y_train,stratify=y_train,
test_size=perc+0.1)
    return X_train, X_test , X_val , y_train , y_val , y_test
def train_valid_test_split_clv(df,perc,target):
    X = df.loc[:, df.columns != target]
    y = df[target]
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=perc)
    X_train, X_val, y_train, y_val  = train_test_split(X_train, y_train, test_size=perc+0.1)
```

```
    return X_train, X_test , X_val , y_train , y_val , y_test
```

**Κανονικοποίηση**

```
def input_normaliser(X,Y,Z,K):
    scaler=StandardScaler()
    scaler.fit(X)
    train=scaler.transform(X)
    valid=scaler.transform(Y)
    test=scaler.transform(Z)
    predict=scaler.transform(K)
    train=pd.DataFrame(train)
    valid=pd.DataFrame(valid)
    test=pd.DataFrame(test)
    predict=pd.DataFrame(predict)
    train.columns=X.columns
    valid.columns=X.columns
    test.columns=X.columns
    predict.columns=X.columns
    train.index=X.index
    valid.index=Y.index
    test.index=Z.index
    predict.index=K.index
    return train , valid , test , predict
```

**Πίνακας 5.11**

```
# CLV correlations

cor = np.abs(customers_train.drop('Customer_no',axis=1).corr())

cor['CLV'].sort_values(ascending=False)
```

**Πίνακας 5.12**

```
# Creating a Random Forest to see the feature importances

random_forest_importances_CLV(customers_train.drop('churn',axis=1),'CLV')
```

**Σχήμα 5.6**

```
plt.hist(np.log1p(X['CLV']),bins=15);
```

**Πίνακας 5.13**

```
product_freq['freq'].describe()
```

**Ανάλυση Κύριων Συνιστωσών**

```
def pca_transformer(X,Y,Z,K,comp):

    pca = PCA(n_components=comp)

    train = pca.fit_transform(X)
```

```
val = pca.transform(Y)

test = pca.transform(Z)

predict = pca.transform(K)


train = pd.DataFrame(data = train
                     , columns = [
                        'pc_'+str(x+1) for x in range(comp)
                     ]
                     , index = X.index)
val = pd.DataFrame(data = val
                     , columns = [
                        'pc_'+str(x+1) for x in range(comp)
                     ]
                     , index = Y.index)
test = pd.DataFrame(data = test
                     , columns = [
                        'pc_'+str(x+1) for x in range(comp)
                     ]
                     , index = Z.index)
predict = pd.DataFrame(data = predict
                     , columns = [
                        'pc_'+str(x+1) for x in range(comp)
                     ]
```

```
                    , index = K.index)
    return train , val , test , predict
```

**Singular Value Decomposition**

```
def svd_transformer(X,Y,Z,comp):
    svd = TruncatedSVD(n_components=comp,n_iter=7,random_state=42)
    train = svd.fit_transform(X)
    val = svd.transform(Y)
    test = svd.transform(Z)


    train = pd.DataFrame(data = train
                    , columns = [
                        'svd_'+str(x+1) for x in range(comp)
                    ]
                    , index = X.index)
    val = pd.DataFrame(data = val
                    , columns = [
                        'svd_'+str(x+1) for x in range(comp)
                    ]
                    , index = Y.index)
    test = pd.DataFrame(data = test
                    , columns = [
                        'svd_'+str(x+1) for x in range(comp)
```

```
                    ]
                    , index = Z.index)
    return train, val, test
```

**Πίνακας 5.14**

```
roc_auc_score(y_val,pred_val)
roc_auc_score(y_ val,pred_ val _xgb)
roc_auc_score(y_ val,prob4)
roc_auc_score(y_val,val_rf)
```

**Σχήμα 5.7**

```
fraction_of_positives, mean_predicted_value = calibration_curve(y_lr, probs, n_bins=10)
fig, ax = plt.subplots(1, figsize=(12, 6))
plt.plot(mean_predicted_value, fraction_of_positives, 's-')
plt.plot([0, 1], [0, 1], '--', color='gray')
sns.despine(left=True, bottom=True)
plt.gca().xaxis.set_ticks_position('none')
plt.gca().yaxis.set_ticks_position('none')
plt.title("$Logistic$ Calibration Curve", fontsize=20)
```

**Σχήμα 5.8**

```
xgb.plot_importance(model_xgb)
```

**Συσχέτιση κύριων συνιστωσών με αρχικές μεταβλητές**

```
for f in X.columns:
    print(f,pearsonr(X_lr['pc_5'],X[f])[0])
```

**Σχήμα 5.9**

```
# Scatter plot of predicted vs real values
plt.scatter(np.expm1(pred_test2),np.expm1(y_test))
plt.plot([0, np.expm1(y_test).max()], [0, np.expm1(y_test).max()], '--', color='gray')
plt.xlabel('Προβλέψεις')
plt.ylabel('Πραγματικές Τιμές')
plt.show()
```

**Σχήμα 5.10**

```
lgb.plot_importance(model)
```

**Πίνακας 5.23**

```
prob3 = model_log.predict_proba(X_pred)
customers_predict['probs'] = prob3
customers_predict[['mean_btv','last_purchase','probs']].sort_values(by='probs',ascending=Fals
e).head(30)
```

**Σχήμα 5.11**

```
preds = np.expm1(model.predict(X_pred, num_iteration=model.best_iteration))
plt.hist(preds);
```

**Πίνακας 5.24**

```
pd.DataFrame(preds).describe()
```

**Μοντέλα**

```
# XGBoost Classifier
def run_xgb(train_X, train_y, val_X, val_y, test_X, test_y):
    params = {'objective': 'binary:logistic',
          'eval_metric': 'logloss',
          'eta': 0.001,
          'max_depth': 4,
          'alpha':1,
          'random_state': 42,
          'silent': True}
    tr_data = xgb.DMatrix(train_X, train_y)
    va_data = xgb.DMatrix(val_X, val_y)
    watchlist = [(tr_data, 'train'), (va_data, 'valid')]
    model_xgb = xgb.train(params, tr_data, 15000, watchlist, maximize=False,
early_stopping_rounds = 100, verbose_eval=100)
    dtest = xgb.DMatrix(test_X)
    xgb_pred_y = model_xgb.predict(dtest, ntree_limit=model_xgb.best_ntree_limit)
    return xgb_pred_y, model_xgb
    # LightGBM Classifier
def run_lgb(train_X, train_y, val_X, val_y, test_X):
    params = {
```

```
        "objective" : "binary",
        "metric" : "binary_logloss",
        "num_leaves" : 40,
        "learning_rate" : 0.001,
        "bagging_fraction" : 0.6,
        "feature_fraction" : 0.6,
        "bagging_seed" : 42,
        "seed": 42
    }
    lgtrain = lgb.Dataset(train_X, label=train_y)
    lgval = lgb.Dataset(val_X, label=val_y)
    evals_result = {}
    model = lgb.train(params, lgtrain, 15000,
                      valid_sets=[lgtrain, lgval],
                      early_stopping_rounds=100,
                      verbose_eval=150,
                      evals_result=evals_result)
    pred_test_y = model.predict(test_X, num_iteration=model.best_iteration)
    return pred_test_y, model, evals_result
# XGBoost Regressor
def run_xgb_reg(train_X, train_y, val_X, val_y, test_X, test_y):
    params = {'objective': 'reg:linear',
          'eval_metric': 'rmse',
```

```
        'eta': 0.001,
        'max_depth': 4,
        'alpha':1,
        'random_state': 42,
        'silent': True}
    tr_data = xgb.DMatrix(train_X, train_y)
    va_data = xgb.DMatrix(val_X, val_y)
    watchlist = [(tr_data, 'train'), (va_data, 'valid')]
    model_xgb = xgb.train(params, tr_data, 15000, watchlist, maximize=False,
early_stopping_rounds = 100, verbose_eval=100)
    dtest = xgb.DMatrix(test_X)
    xgb_pred_y = model_xgb.predict(dtest, ntree_limit=model_xgb.best_ntree_limit)
    return xgb_pred_y, model_xgb
# LightGBM Regressor


def run_lgb_reg(train_X, train_y, val_X, val_y, test_X):
    params = {
        "objective" : "regression",
        "metric" : "rmse",
        "num_leaves" : 40,
        "learning_rate" : 0.001,
        "bagging_fraction" : 0.6,
        "feature_fraction" : 0.6,
        "seed": 42
```

```
    }
    lgtrain = lgb.Dataset(train_X, label=train_y)
    lgval = lgb.Dataset(val_X, label=val_y)
    evals_result = {}
    model = lgb.train(params, lgtrain, 15000,
                      valid_sets=[lgtrain, lgval],
                      early_stopping_rounds=100,
                      verbose_eval=150,
                      evals_result=evals_result)
    pred_test_y = model.predict(test_X, num_iteration=model.best_iteration)
    return pred_test_y, model, evals_result
reg = l1_l2(l1=0, l2=0.01)
model_log = Sequential()
model_log.add(Dense(1, activation='sigmoid', W_regularizer=reg,
input_dim=X_train.shape[1]))
model_log.compile(optimizer='Adam', loss='binary_crossentropy')
model_log.fit(X_train, y_train, nb_epoch=700, validation_data=(X_val, y_val))
# Number of trees in random forest
n_estimators = [int(x) for x in np.linspace(start = 200, stop = 2000, num = 10)]
# Number of features to consider at every split
max_features = ['auto', 'sqrt']
# Maximum number of levels in tree
max_depth = [int(x) for x in np.linspace(5, 20, num = 11)]
max_depth.append(None)
```

```
# Minimum number of samples required to split a node
min_samples_split = [2, 5, 10]
# Minimum number of samples required at each leaf node
min_samples_leaf = [1, 2, 4]
# Method of selecting samples for training each tree
bootstrap = [True, False]
# Create the random grid
random_grid = {'n_estimators': n_estimators,
               'max_features': max_features,
               'max_depth': max_depth,
               'min_samples_split': min_samples_split,
               'min_samples_leaf': min_samples_leaf,
               'bootstrap': bootstrap}
cv = StratifiedKFold(n_splits=5)
rf = RandomForestClassifier()
rf_random = RandomizedSearchCV(estimator = rf,
                               param_distributions = random_grid
                               , n_iter = 100
                               , cv = cv
                               , verbose=2
                               , random_state=42
                               , n_jobs = -1)
```

# Βιβλιογραφία


[1] Κούτρας Μ.Β. (2004). Εφαρμοσμένη Πολυμεταβλητή Ανάλυση, Πανεπιστημιακές Σημειώσεις, ΠΜΣ «Εφαρμοσμένη Στατιστική».

[2] Καρλής Δ. (2005). Πολυμεταβλητή Στατιστική Ανάλυση, Εκδόσεις Σταμούλη, Αθήνα

[3] Trevor Hastie, Robert Tibshirani and Jerome Friedman (2008). The Elements of Statistical Learning

[4] Ali Vanderveld, Addhyan Pandey, Angela Han, and Rajesh Parekh. 2016. An Engagement-Based Customer Lifetime Value System for E-commerce. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.

[5] Bianca Zadrozny and Charles Elkan. 2001. Obtaining Calibrated Probability Estimates from Decision Trees and Naive Bayesian Classifiers. Proceedings of the 18th International Conference on Machine Learning 1 (2001), 609–616.

[6] Aditya Krishna Menon, Xiaoqian Jiang, Shankar Vembu, Charles Elkan and Lucila Ohno-Machado (2012). Predicting accurate probabilities with a ranking loss.

[7] Alexandru Niculescu-Mizil and Rich Caruana (2005). Predicting Good Probabilities With Supervised Learning.

[8] Tianqi Chen and Carlos Guestrin (2016). XGBoost: A Scalable Tree Boosting System

[9] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, , Weidong Ma, Qiwei Ye and Tie-Yan Liu (2017). LightGBM: A Highly Efficient Gradient Boosting Decision Tree.

[10] Benjamin Paul Chamberlain, Angelo Cardoso and C.H. Bryan Liu (2017). Customer Lifetime Value Prediction Using Embeddings.

[11] Makoto, A. (2008). Counting your Customers one by one: a hierarchical Bayes extension to the Pareto/NBD model, Marketing Science.

[12] Everitt, B. S. and Dunn, G. (1991). Applied Multivariate Data Analysis, Arnold, New York.

[13] Berger, P. D. and Nasr, N. I. (1998). Customer lifetime value: Marketing models and applications. Journal of Interactive Marketing, 12, 17–30.

[14] Fader, P. S., Hardie, G. S. and Lee, K. L. (2005). Counting your customers the easy way: An alternative to the Pareto/NBD model. Marketing Science, 24, 275–284.

[15] Toon Calders and Szymon Jaroszewicz (2007). Efficient AUC Optimization for Classification

[16] Franses, Philip Hans and Richard Paap (2001). Quantitative Models in Marketing Research, Cambridge University Press, Cambridge, UK.

[17] Jerome H. Friedman (2001). Greedy Function Approximation A Gradient Boosting Machine.

[18] Leo Breiman (2001). Random Forests.

[19] Su-In Lee, Honglak Lee, Pieter Abbeel and Andrew Y. Ng (2006). Efficient L1 Regularized Logistic Regression.

[20] Gennady G. Pekhimenko (2013). Penalizied Logistic Regression for Classification.

[21] Oren Barkan and Noam Koenigstein (2016). Item2Vec: Neural Item Embedding for Collaborative Filtering

[22] Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall and W. Philip Kegelmeyer (2002). SMOTE: Synthetic Minority Over-sampling Technique, Journal of Artificial Intelligence Research 16.