Πανεπιστήμιο Πελοποννήσου

Σχολή Μηχανικών

Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Πτυχιακή Εργασία

Συλλογή και διερευνητική ανάλυση στατιστικών δεδομένων αγώνων διεθνών τουρνουά αντισφαίρισης

Φοιτητής: Βακόνδιος Μάρκος

Αριθμός Μητρώου: 15049

Π. ΜΗΠΤΕ (ΕΠΔΟ-ΜΕΣ)

Επιβλέπων Καθηγητής: Κούτρας Αθανάσιος

21 Ιουνίου 2023

Υπεύθυνη Δήλωση περί μη λογοκλοπής

Βεβαιώνω ότι είμαι συγγραφέας αυτής της εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της, είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, έχω αναφέρει τις όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών ή λέξεων, είτε αυτές αναφέρονται ακριβώς είτε παραφρασμένες. Ακόμα δηλώνω ότι αυτή η γραπτή εργασία προετοιμάστηκε από εμένα προσωπικά και αποκλειστικά και ειδικά για την συγκεκριμένη πτυχιακή εργασία και ότι θα αναλάβω πλήρως τις συνέπειες εάν η εργασία αυτή αποδειχθεί ότι δεν μου ανήκει.

ΟΝΟΜΑΤΕΠΩΝΥΜΟ ΦΟΙΤΗΤΗ 1	**ΑΜ**	**ΥΠΟΓΡΑΦΗ**
Βακόνδιος Μάρκος	15049	

Ευχαριστίες

Θα ήθελα να ευχαριστήσω την οικογένεια μου που με βοήθησε σε κάθε βήμα μέχρι τώρα. Επίσης, θα ήθελα να ευχαριστήσω τον κ.Κούτρα για την βοήθεια στην πορεία της συγγραφής.

Πρόλογος

Αντικείμενο της πτυχιακής εργασίας είναι η διερευνητική ανάλυση (exploratory data analysis) και η απεικόνιση των αποτελεσμάτων ανάλυσης μεγάλου αριθμού δεδομένων βασισμένα σε στατιστικά στοιχεία διεθνών τουρνουά αντισφαίρισης. Έγινε χρήση σε πακέτα ανοικτού κώδικα στατιστικής επεξεργασίας μεγάλων δεδομένων τα οποία βασίζονται στην γλώσσα python για την εύρεση πληροφορίας και την ανάδειξη της μέσω της χρήσης βιβλιοθηκών απεικόνισης σε γλώσσα python.

Περίληψη

Η πτυχιακή αυτή πραγματεύεται την ανάλυση αγώνων επαγγελματικού τέννις, με απότερο στόχο την πρόβλεψη του αποτελέσματος αυτών, με βάση τα στατιστικά των αγώνων και τα χαρακτηριστικά των παικτών κατά τη διάρκεια μίας μεγάλης χρονικής περιόδου.

Το σύνολο δεδομένων που αναλύθηκαν πραγματεύεται την πορεία των καλύτερων επαγγελματιών παικτών, με βάση το ranking τους, μεταξύ 2000 και 2017 και τα δεδομένα έχουν εξαχθεί από το τουρ της Association of Tennis Professionals (ATP), η οποία είναι η διεθνής ομοσπονδία τέννις. Στο σύνολο των αγώνων αυτών που αναλύθηκαν ήταν τα κύρια τουρνουά (Grand Slams) καθώς και αυτά τα οποία θεωρούνται τα σημαντικότερα για την κατάταξη των παιχτών, κατά τη διάρκεια κάθε έτους.

Με την χρήση του Exploratory Data Analysis (EDA), ήταν δυνατή η ανάλυση αυτού του dataset με ένα αποδοτικό τρόπο, ο οποίος βοήθησε στην εξαγωγή συμπερασμάτων, τα οποία μπορούν να οδηγήσουν και σε προβλέψεις για τα αποτελέσματα των αγώνων.

Αφού αναλύθηκαν οι διάφοροι τύποι ανάλυσης, οι τεχνικές και τα εργαλεία που χρησιμοποιούνται στη διεθνή βιβλιογραφία, έγινε εφαρμογή της ανάλυσης στο dataset που είχαμε διαθέσιμο.

Έτσι, για παράδειγμα, με τη χρήση των εργαλείων και της γλώσσας προγραμματισμού Python βρήκαμε αντιπάλους οι οποίοι δυσκόλευαν ιδιαίτερα συγκεκριμένους παίχτες, χωρίς αυτό να αντικατοπτρίζεται από την κατάταξή τους.

Τα αποτελέσματα της ανάλυσης συνάδουν με ορισμένες εργασίες που βρέθηκαν και εφαρμόστηκαν στη διεθνή βιβλιογραφία, με τη διαφορά ότι η οπτική στην ανάλυση των μεταβλητών δεν βρέθηκε αυτούσια στη διεθνή βιβλιογραφία (π.χ. μεταβλητές).

Abstract

The objective of this thesis was the analysis of professional tennis games, having as a target the prediction of their result. The basis was the statistics that were exported from the games, as well as characteristics of the players within a period of time.

The Data set that was analysed had the results of the games of the best professional tennis players, based on their ranking, between 2000 and 2017. The data has been extracted from the tour of the Association of Tennis Professionals (ATP), which is in fact the international association. The games analysed were the main tournaments (Grand Slams), as well as the tournaments that are the most important for the ranking of the players, during each year.

By Using Exploratory Data Analysis (EDA), it was possible to analyse the data set in an effective way, that allowed us to extract conclusions. These conclusions can lead to predictions of the results of the games.

For the thesis, a number of analysis types were presented, techniques, as well as tools that are used in the international bibliography. After that, the data set was used to apply some of these techniques.

So for example, by using tools and the Python programming language, we found opponents that were tough to beat for specific players, something that was not represented in the rankings.

The results of this analysis are aligned with some references that were found and applied in the international bibliography. There was a main difference in the analysis and the objectives of this thesis, compared to the international bibliography and that was the view that was different (e.g. variables).

Περιεχόμενα

Κατάλογος σχημάτων

Κατάλογος πινάκων

Εισαγωγή

Οι άνθρωποι, κατά τη διάρκεια της ιστορίας, προσπάθησαν να προβλέψουν αποτελέσματα αθλητικών γεγονότων. Υπάρχουν αναφορές ότι κατά την περίοδο της αρχαίας Ελλάδας ακόμη, άνθρωποι χρησιμοποιούσαν τα μαντεία ώστε να προβλεφθεί η επιτυχία αθλητών στους ολυμπιακούς αγώνες. [22] [21]

Αυτή η επιθυμία της πρόβλεψης του μέλλοντος στο περιβάλλον των αθλημάτων, συνεχίζεται έως και σήμερα. Έχοντας ως αντικείμενο αθλήματα, αλλά και πτυχές της ζωής, η αγορά στοιχημάτων είναι μία βιομηχανία δισσεκατομυρίων. Την εποχή του διαδικτύου, η αγορά μεταφέρθηκε σε ηλεκτρονική μορφή, με ιδιαίτερα υψηλά κέρδη για τις εταιρείες αφού πλέον υπάρχει μεγαλύτερη προσβασιμότητα για τους παίκτες και συνεπώς, μεγαλύτερη ευκολία στην τοποθέτηση στοιχημάτων.

Ως άθλημα, το τέννις είναι δίχως αμφιβολία, ένα από τα πιο δημοφιλή αθλήματα στον κόσμο. Εκατομμύρια άνθρωποι ανά την υφήλιο αθλούνται και ορισμένα από τα τουρνουά που διοργανώνονται είναι υψηλού κύρους και συγκεντρώνουν ένα σημαντικό μέρος ενδιαφέροντος από τη διεθνή κοινότητα. Το τέννις είναι ένα πολύ ενδιαφέρον θέμα και απασχολεί και τη διεθνή ερευνητική κοινότητα, καθώς υπάρχει υψηλή φήμη και δημοτικότητα στο άθλημα. Επιπλέον, η πιθανότητα της μετατροπής των γνώσεων σε κέρδος, στρέφει το ενδιαφέρον και της στοιχηματικής κοινότητας πάνω στο άθλημα.

Ο στόχος της πτυχιακής εργασίας αυτής είναι να δημιουργηθεί ένα σύστημα το οποίο θα αναλύει τα στατιστικά των ανδρών επαγγελματιών παικτών τέννις για τα έτη 2000-2017. Με τη χρήση εργαλείων, καθώς και των πληροφοριών που δίνονται από το dataset για τους αγώνες τέννις, θέλουμε να αναλύσουμε τα δεδομένα και να βγάλουμε συμπεράσματα τα οποία θα μπορούσαν να χρησιμοποιηθούν για την εξαγωγή προβλέψεων, και τα οποία δεν μπορούν να εξαχθούν με ευκολία. Έτσι, ιδανικά, θα θέλαμε να προβλέψουμε την πιθανότητα νίκης κάθε παίχτη σε μελλοντικό αγώνα σε μία απόπειρα να μετατρέψουμε σε κέρδος αυτή την πληροφορία στην αγορά του online στοιχήματος.

Όπως και η πλειονότητα των πηγών, το μεγαλύτερο κομμάτι της έρευνας επικεντρώθηκε

στην ανάλυση των επαγγελματιών παικτών της ATP, για τα μονά παιχνίδια, στο τουρ που διοργανώνεται κάθε έτος. Με τον όρο τουρ εννοούμε τα μεγαλύτερα τουρνουά, τα οποία αντιστοιχούν στα grand slams, καθώς και τα ATP masters των 1000, 500 και 250 πόντων αντίστοιχα.

0.1 Παρόμοιες εργασίες

Στο κομμάτι των προβλέψεων υπάρχουν πάρα πολλές εργασίες και βιβλιοθήκες [5] που πραγματεύονται το κομμάτι των προβλέψεων για διάφορα αθλήματα. Εμείς θα επικεντρωθούμε στο κομμάτι των προβλέψεων στο τέννις, καθώς είναι και το κύριο ενδιαφέρον αυτής της εργασίας.

Στο κομμάτι των βιβλιοθηκών, λόγω της ευκολίας της python, υπάρχουν διάφορα πακέτα τα οποία είναι ανοιχτού κώδικα, και έχουν να κάνουν με την ανάλυση δεδομένων για αθλήματα. Το πιο ενδιαφέρον στην περίπτωσή μας είναι το tennisim[5]. Το πακέτο αυτό μπορεί να χρησιμοποιηθεί για την προσομοίωση των αγώνων τέννις με μοντελοποίηση με βάση τους πόντους, ώστε να δοθεί η πιθανότητα νίκης ενός παίχτη σε ένα πόντο. Έτσι, μπορεί να γίνει προσομοίωση και παιχνιδιών, καθώς και κάθε πιθανού αποτελέσματος (πόντοι, games, κ.ά), καθώς και των παραμέτρων που μπορεί να υπάρχει ενδιαφέρον και να αντιληφθούμε έτσι τι μπορεί να έχει σημασία στο παιχνίδι ενός παίχτη.

Ως προς το ερευνητικό μέρος, μία περίπτωση είναι η έρευνα [20], κατά την οποία προτείνεται μία στατιστική προσέγγιση για την πρόβλέψη των αποτελεσμάτων των grand slam αγώνων με τη χρήση EDA. Η προσέγγιση δημιουργεί επιπλέον μεταβλητές, με τη χρήση της μεθόδου του Bayes. Τα αποτελέσματα έδειξαν για την ανάλυση ότι το ranking είναι η σημαντικότερη μεταβλητή και ότι το μοντέλο που προτάθηκε έδεινε καλύτερη ακρίβεια σε σύγκριση με άλλα μοντέλα.

Μία άλλη ερευνητική πρόταση ήταν η [16] κατά την οποία οι ερευνητές προτείνουν μία προσέγγιση με τη χρήση τεχνητής νοημοσύνης για την πρόβλεψη του νικητή των αγώνων του ATP tour στους αγώνες των ανδρών με ακρίβεια. Με τη χρήση ενός σετ δεδομένων ανοιχτού κώδικα, με ιστορικά δεδομένα από όλα τα επίπεδα, εξήχθησαν 84 μεταβλητές με τη σημαντικότητα να ορίζεται με βάση προηγούμενη έρευνα. Το μοντέλο του logistic regression ήταν καλύτερο σε πρόβλεψη σε σχέση με αυτό του επίσημου ATP-ranking.

Μία παρόμοια προσέγγιση ήταν και στην περίπτωση του [17], κατά την οποία γίνεται χρήση ιστορικών στοιχείων για την εξαγωγή 22 μεταβλητών από ιστορικά δεδομένα. (Μερικά

από αυτά ήταν η κούραση και οι τραυματισμοί, που διαφοροποίησαν την ανάλυση, σύμφωνα με το συγγραφέα.) Με τη χρήση των δεδομένων δημιουργείται και γίνεται βελτιοστοποίση σε διάφοα μοντέλα, όπως του logistic regression, καθώς και των νευρωνικών δικτύων. Όταν έγινε σύγκριση σε ένα αριθμό 6315 αγώνων ATP κατά τη σεζόν 2013-2014, το μοντέλο του συγγραφέα έδωσε καλύτερα ποσοστά ακρίβειας σε σύγκριση με το μοντέλο Knottenbelt's Common-Opponent model, που είναι το πιο ακριβές στη βιβλιογραφία.

Τέλος, ένα ακόμη μοντέλο που δημιουργήθηκε για το χαρακτηρισμό και την πρόβλεψη της επιτυχίας των επαγγελματιών αγώνων τέννις ήταν το [19], κατά το οποίο δόθηκε η απόδοση σε 40 αγώνες. Η πιθανοτική πρόβλεψη επιτυχίας των παικτών μετατρέπεται σε ένα πρόβλημα δυαδικής κατηγοριοποίησης, που επιλύθηκε και εδώ με τη χρήση Logistic Regression και Νευρωνικών δικτύων.

Κεφάλαιο 1

Exploratory Data Analysis (EDA)

1.1 Χρήση του EDA

Είναι δυνατό, με τη χρήση του Exploratory data analysis να βελτιωθούν ορισμένες αποφάσεις και να αποκτηθούν γνώσεις, με την εφαρμογή του στο τέννις.

Το EDA λοιπόν, μπορεί να βοηθήσει αθλητές, επαγγελματίες, και μη, με τους εξής τρόπους:

- **Ανάλυση της απόδοσης των παικτών:** Με τη χρήση του EDA, μπορεί να γίνει ανάλυση για ένα παίκτη σε βάθος χρόνου, συλλέγοντας και αναλύοντας πληροφορίες που έχουν να κάνουν με το σερβίς, τα ποσοστά νικών του κ.ά. Προπονητές και παίκτες μπορούν να βελτιώσουν τομείς του παιχνιδιού που χρήζει βελτίωσης, προσαρμόζοντας την προπόνηση. Για παράδειγμα, εάν το ποσοστό επιτυχίας έναντι σε αντιπάλους με αριστερό χέρι είναι χαμηλό, θα πρέπει να ακολουθηθεί ειδική προπόνηση.

- **Ανάλυση αγώνων:** Οι προπονητές και οι παίκτες μέσω του EDA μπορούν να δημιουργήσουν στρατηγικές για αγώνες αναλύοντας δεδομένα σχετικά με ένα αντίπαλο. Ένας παίκτης που κάνει πολλά λάθη σε επιφάνεια χόρτου μπορεί να έχει μειονέκτημα εάν αυτό ανακαλυφθεί και έτσι το παιχνίδι μπορεί να αλλάξει με στόχο να εξαναγκαστούν περισσότερα λάθη. Το ίδιο μπορεί να συμβεί και για ένα παίχτη, και μέσω του EDA να γνωρίζει τα δυνατά μέρη του παιχνιδιού του αντιπάλου του.

- **Έγκαιρη Πρόληψη τραυματισμών:** Δεδομένα τραυματισμών μπορούν επίσης να αναλυθούν με τη χρήση του EDA, και να οδηγήσουν στην ανακάλυψη μοτίβων. Αυτά μπορεί να έχουν να κάνουν με τον τύπο τραυματισμού, τα στοιχεία του παίκτη, τη συχνότητα ή και τον τύπο τραυματισμού. Έτσι μπορούν να αναπτυχθούν ειδικά προ-

γράμματα προπόνησης για την πρόληψη τραυματισμών, ακόμη και κατά τη διάρκεια του έτους.

- **Αλληλεπίδραση με θαυμαστές:** Με τη χρήση του EDA, μπορούμε να έχουμε αύξηση του engagment στο τέννις. Μέσω της ανάλυσης δεδομένων πωλήσεων, δραστήριοτήτων στα social media καθώς και της τηλεθέασης, μοτίβα μπορούν να εντοπιστούν από τους διοργανωτές. Έτσι, οι θεατές μπορούν να έχουν μία αυξημένη εμπειρία μέσω του εξατομικευμένου μάρκετινγκ και των δράσεων που μπορεί να υπάρξουν κατά τη διάρκεια των αγώνων.

- **Ανάλυση εξοπλισμού:** Ένα ακόμη μέρος που μπορεί να χρησιμοποηθεί το EDA είναι αυτό του εξοπλισμού. Με τη χρήση δεδομένων τα οποία ενίοτε μπορεί να απαιτούν ειδικές συνθήκες (εργαστηρίου), μπορούν να αναπτυχθούν προϊόντα με βάση τις ανάγκες των αθλητών. Για παράδειγμα, η γκάμα των ρακετών μίας εταιρείας μπορεί να ανταποκρίνεται στις ανάγκες αθλητών, με τη χρήση συγκεκριμένης κεφαλής. Έτσι, η εταιρεία αυτή είναι πιο πιθανό να επιτύχει και να έχει κέρδος για αυτό το προϊόν, διότι οι αθλητές θα λύνουν ένα πρόβλημα που μπορεί να έχουν με άλλες ρακέτες.

Εϊναι κατανοητό συνεπώς, ότι το EDA έχει προοπτικές για την εφαρμογή του από διάφορους ανθρώπους του παιχνιδιού (παίκτες, προπονητές, διοργανωτές, κατασκευαστές). Το EDA μπορεί να βοηθήσει κάθε ένα στη λήψη αποφάσεων, οι οποίες είναι πλέον data-driven. Ως εκ τούτου, οι αποφάσεις αυτές είναι πιο πιθανό να οδηγήσουν σε βέλτιση απόδοση (παιχνιδιού, προϊόντων κ.ο.κ.).

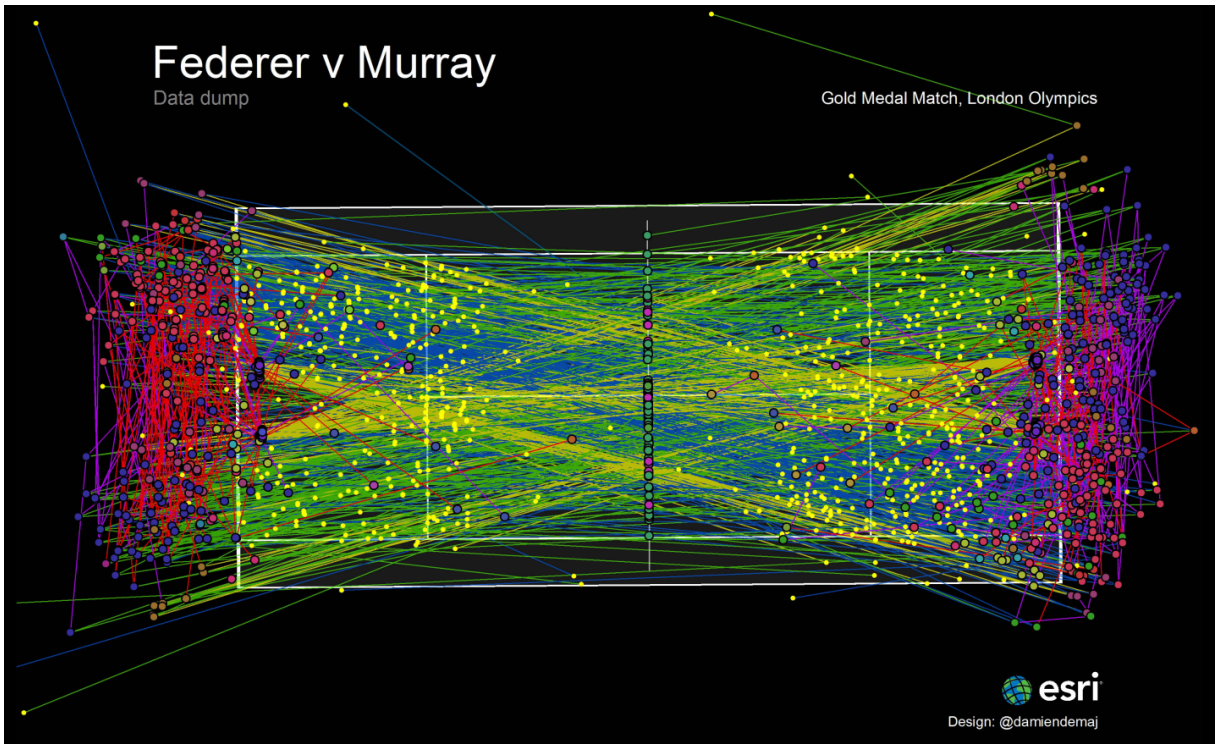
1.2 Ορισμοί (Exploratory data analysis, terms, etc.)

Τα δεδομένα μπορεί να αναπαριστούν διακριτά αντικείμενα, αριθμούς, φράσεις, γεγονότα, μετρήσεις, παρατηρήσεις, ακόμη και περιγραφές πραγμάτων. Κάθε γεγονός ή διαδικασία σε διάφορους τομείς, συμπεριλαμβανομένης της βιολογίας, της οικονομίας, της μηχανικής και του μάρκετινγκ, συλλέγει και αποθηκεύει τέτοιες πληροφορίες.[9]

Με την επεξεργασία αυτών των δεδομένων μπορούν να αποδοθούν πολύτιμες πληροφορίες και η επεξεργασία αυτών των πληροφοριών αποδίδει ωφέλιμη γνώση. Σημαντική διαδικασία, είναι ο τρόπος εξαγωγής σημαντικών και χρηστικών πληροφοριών από τέτοια δεδομένα. Η EDA είναι η λύση σε αυτό το ερώτημα, καθώς η EDA είναι η διαδικασία ανάλυσης

προσβάσιμων συνόλων δεδομένων με σκοπό τον εντοπισμό προτύπων, καθώς τον εντοπισμό ανωμαλιών, τον έλεγχο υποθέσεων και την επικύρωση υποθέσεων χρησιμοποιώντας στατιστικές μετρήσεις. Σε αυτό το κεφάλαιο, θα καλύψουμε τις διαδικασίες που απαιτούνται για να κάνουμε διερευνητική ανάλυση δεδομένων.

Ο βασικός στόχος της EDA είναι να προσδιορίσει τι δεδομένα μπορούν να χρησιμοποιηθούν πριν από την επίσημη μοντελοποίηση ή τη δημιουργία υποθέσεων. Ο John W. Tukey [18] ενθάρρυνε τους στατιστικολόγους να χρησιμοποιήσουν την EDA για να διερευνήσουν και να αποκαλύψουν δεδομένα και να δημιουργήσουν νέες υποθέσεις που θα μπορούσαν να χρησιμοποιηθούν για τη δημιουργία μιας νέας μεθόδου συλλογής δεδομένων και πειραμάτων [13].



Σχήμα 1.1: Όλα τα σουτ του τελικού των ολυμπιακών αγώνων 2012

Στην εικόνα 1.1 παρατηρούμε όλα τα σουτ κατά τον τελικό των ολυμπιακών αγώνων του Λονδίνου. Η εικόνα δείχνει να μην είναι ξεκάθαρη, και έτσι δυσκολευόμαστε να βγάλουμε κάποιο συμπέρασμα. Με τη χρήση του EDA, μπορούμε να εξάγουμε συμπεράσματα μέσα από την ανάλυση των μεταβλητών αυτών, και έτσι να κάνουμε προβλέψεις σχετικά με αγώνες στο μέλλον.

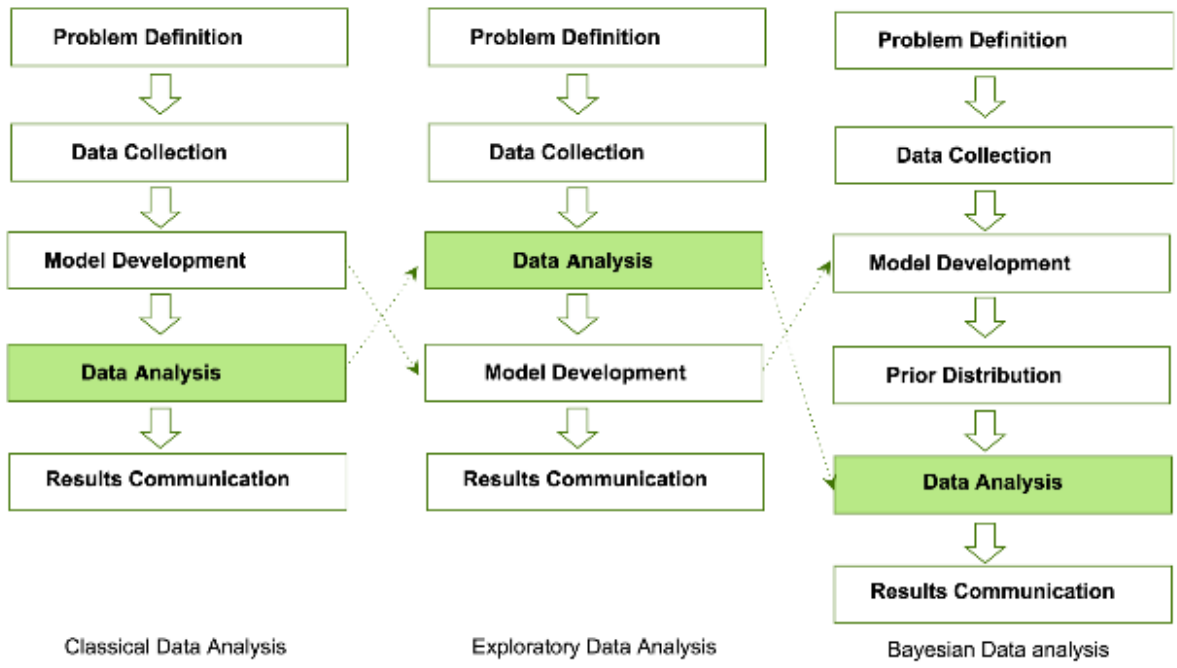
1.3 Σύγκριση EDA με κλασική και Bayesian ανάλυση

Υπάρχουν διάφορες μέθοδοι ανάλυσης δεδομένων. Τα ακόλουθα είναι τα πιο δημοφιλή [8] [2]:

- Κλασική ανάλυση δεδομένων: Στην κλασική μέθοδο ανάλυσης δεδομένων, η διατύπωση προβλημάτων και η συλλογή δεδομένων ακολουθούνται από τη δημιουργία μοντέλου, την ανάλυση και την επικοινωνία αποτελεσμάτων.

- Όπως ήδη έχει αναφερθεί, η διερευνητική ανάλυση δεδομένων (EDA) ακολουθεί τα ίδια στάδια με την παραδοσιακή ανάλυση δεδομένων, διαφέρει μονό στα βήματα επιβολής του μοντέλου και της ανάλυσης δεδομένων όπου και αντιστρέφονται. Τα δεδομένα, η δομή τους, τα ακραία σημεία, τα μοντέλα και οι οπτικοποιήσεις είναι η κύρια εστίαση. Γενικά, τα ντετερμινιστικά ή πιθανολογικά μοντέλα δεν επιβάλλονται στα δεδομένα στο EDA.

- Τεχνική ανάλυσης δεδομένων Bayesian: Όπως φαίνεται στο συνημμένο σχήμα, η Bayesian προσέγγιση προσθέτει προηγούμενη γνώση κατανομής πιθανοτήτων στις διαδικασίες ανάλυσης. Με λίγα λόγια, η προηγούμενη κατανομή πιθανοτήτων οποιουδήποτε αριθμού μεταφέρει την πεποίθηση για αυτήν την ποσότητα πριν από την εξέταση αποδεικτικών στοιχείων. Το παρακάτω γράφημα απεικονίζει τρεις διαφορετικές τεχνικές ανάλυσης δεδομένων, επισημαίνοντας τις διαφορές μεταξύ τους στην υλοποίηση.



Σχήμα 1.2: Τύποι ανάλυσης δεδομένων

Οι αναλυτές δεδομένων και οι επιστήμονες δεδομένων μπορούν ελεύθερα να συνδυάσουν τις τεχνικές που περιγράφονται στις προηγούμενες μεθόδους για να εξαγάγουν σχετικές γνώσεις από τα δεδομένα. Επιπλέον, είναι δύσκολο να προσδιοριστεί ή να προβλεφθεί ποιο μοντέλο ανάλυσης δεδομένων είναι το καλύτερο. Κάθε ένα έχει τη δική του εφαρμογή και είναι κατάλληλο για διάφορες μορφές επεξεργασίας δεδομένων.

1.4 Βήματα στην EDA

Αφού κατανοήσαμε τι είναι η EDA και τη συνάφειά της, ας εξετάσουμε τις διάφορες διαδικασίες ανάλυσης δεδομένων. Ουσιαστικά, εμπλέκονται τέσσερις διακριτές φάσεις. Ας εξετάσουμε την καθεμία για να αποκτήσουμε βασικές γνώσεις για κάθε διαδικασία [18] [2] [3].

- **Ορισμός Προβλήματος:** Πριν από την προσπάθεια εξαγωγής σχετικών πληροφοριών από δεδομένα, είναι σημαντικό να περιγραφεί το επιχειρηματικό πρόβλημα που πρέπει να επιλυθεί. Ο ορισμός του προβλήματος είναι η κινητήρια δύναμη πίσω από την εκτέλεση ενός σχεδίου ανάλυσης δεδομένων. Ο προσδιορισμός του κύριου σκοπού της ανάλυσης, ο καθορισμός των βασικών παραδοτέων, η λεπτομέρεια των κύριων ρόλων και ευθυνών, η λήψη της τρέχουσας κατάστασης των δεδομένων, ο καθορισμός του

χρονοδιαγράμματος και η ολοκλήρωση μιας ανάλυσης κόστους-οφέλους είναι οι κύριες δραστηριότητες που εμπλέκονται στον ορισμό του ζητήματος. Με βάση αυτόν τον ορισμό του προβλήματος, μπορεί να αναπτυχθεί ένα σχέδιο δράσης.

- Προετοιμασία δεδομένων: Αυτή η διαδικασία συνεπάγεται την προετοιμασία του συνόλου δεδομένων για ανάλυση. Κατά τη διάρκεια αυτής της φάσης, προσδιορίζουμε τις πηγές δεδομένων, καθιερώνουμε τα σχήματα και τους πίνακες δεδομένων, κατανοούμε τις κύριες ιδιότητες των δεδομένων, καθαρίζουμε το σύνολο δεδομένων, αφαιρούμε μη σχετικά σύνολα δεδομένων, μετατρέπουμε τα δεδομένα και χωρίζουμε τα δεδομένα σε απαιτούμενα κομμάτια για ανάλυση.

- Ανάλυση δεδομένων: Είναι μια από τις πιο σημαντικές διαδικασίες που αφορούν την περιγραφική στατιστική και την ανάλυση δεδομένων. Οι κύριες αρμοδιότητες συνίστανται στην περίληψη των δεδομένων, την ανακάλυψη κρυφών συσχετισμών και συνδέσεων μεταξύ των δεδομένων, την κατασκευή μοντέλων πρόβλεψης, την αξιολόγηση των μοντέλων και την εκτίμηση της ακρίβειάς τους. Συνοπτικοί πίνακες, γραφικά, περιγραφικές στατιστικές, στατιστικές συμπερασμάτων, στατιστικές συσχέτισης, αναζήτηση, ομαδοποίηση και μαθηματικά μοντέλα είναι μερικές από τις προσεγγίσεις που χρησιμοποιούνται για τη σύνοψη δεδομένων.

- Ανάπτυξη και αναπαράσταση αποτελεσμάτων: Αυτή η διαδικασία περιλαμβάνει την παρουσίαση του συνόλου δεδομένων με τη μορφή γραφημάτων, συνοπτικών πινάκων, χαρτών και διαγραμμάτων στο κοινό που θέλουμε. Αυτό είναι επίσης ένα κρίσιμο στάδιο, καθώς ένας από τους πρωταρχικούς στόχους της EDA είναι τα αποτελέσματα της ανάλυσης δεδομένων να είναι ερμηνεύσιμα από τα ενδιαφερόμενα μέρη της διαδικασίας. Η πλειονότητα των μεθόδων γραφικής ανάλυσης αποτελείται από διαγράμματα διασποράς, γραφικές παραστάσεις χαρακτήρων, ιστογράμματα, γραφικές παραστάσεις πλαισίου, υπολειμματικές γραφικές παραστάσεις και διάγραμμα μέσης τιμής, μεταξύ άλλων.

1.5 Εφαρμογές (της EDA)

Ο πρωταρχικός στόχος της EDA είναι να βοηθήσει στην εξέταση των δεδομένων πριν από τη διαμόρφωση οποιονδήποτε υποθέσεων. Μπορεί να βοηθήσει στον εντοπισμό προφανών

λαθών, καθώς και στην καλύτερη κατανόηση των προτύπων (patterns) μέσα στα δεδομένα, στην ανίχνευση ακραίων ή ασυνήθιστων περιστατικών και στην ανακάλυψη σημαντικών σχέσεων μεταξύ των μεταβλητών.[13]

Η διερευνητική ανάλυση μπορεί να χρησιμοποιηθεί από επιστήμονες δεδομένων για να εγγυηθεί ότι τα αποτελέσματα που δημιουργούν είναι σωστά και κατάλληλα για τα στοχευμένα αποτελέσματα και τους στόχους οι οποίοι τίθενται. Η EDA βοηθά επίσης τους ενδιαφερόμενους φορείς διασφαλίζοντας ότι κάνουν τις κατάλληλες ερωτήσεις, ενώ μπορεί να βοηθήσει και στον προσδιορισμό τυπικών αποκλίσεων, κατηγορικών μεταβλητών και διαστημάτων εμπιστοσύνης. Όταν ολοκληρωθεί η EDA και εξαχθούν πληροφορίες, τα χαρακτηριστικά του μπορούν να χρησιμοποιηθούν για μία πιο σύνθετη ανάλυση ή μοντελοποίηση δεδομένων, συμπεριλαμβανομένης της μηχανικής μάθησης [18] [15].

Τα εργαλεία EDA μπορούν να εκτελέσουν τις ακόλουθες στατιστικές λειτουργίες και τεχνικές:

- Τεχνικές για ομαδοποίηση και μείωση διαστάσεων των δεδομένων, που βοηθούν στη δημιουργία γραφικών αναπαραστάσεων δεδομένων υψηλής ποιότητας με πολλές μεταβλητές.

- Κάθε πεδίο ακατέργαστων συνόλων δεδομένων εμφανίζεται με μορφή μονής μεταβλητής , μαζί με συνοπτικά στατιστικά στοιχεία.

- Οπτικοποίηση δύο μεταβλητών και συνοπτικά στατιστικά στοιχεία που επιτρέπουν την αξιολόγηση της σύνδεσης μεταξύ κάθε μεταβλητής στο σύνολο δεδομένων και της μεταβλητής στόχου.

- Απεικονίσεις πολλών μεταβλητών που χρησιμοποιούνται για τη χαρτογράφηση και την ανάλυση των σχέσεων μεταξύ πολλαπλών μεταβλητών στα δεδομένα.

- **K-means Clustering:** Ο αλγόριθμος αυτός χρησιμοποιείται για λύσεις συσταδοποίσης σε unsupervised learning. Δεν υπάρχει κατηγοριοποίηση η label προηγουμένως, και τα αντικείμενα διαιρούνται σε k συστάδες (ομάδες) δεδομένων, τα οποία είναι παρόμοια για ένα χαρακτηριστικό.

- Τα προγνωστικά μοντέλα, όπως η γραμμική παλινδρόμηση (Linear regression), βασίζονται σε στατιστικές και δεδομένα για να κάνουν προβλέψεις.

1.6 Τύποι Διερευνητικής Ανάλυσης Δεδομένων

Η διερευνητική ανάλυση δεδομένων συχνά χωρίζεται σε δύο κατηγορίες.

Αρχικά, κάθε προσέγγιση είναι είτε γραφική είτε μη γραφική. Δεύτερον, κάθε προσέγγιση είναι είτε μονής μεταβλητής είτε πολλών μεταβλητών (συνήθως απλώς δύο). Οι μη γραφικές προσεγγίσεις συχνά περιλαμβάνουν τον υπολογισμό των συνοπτικών στατιστικών, αλλά οι γραφικές μέθοδοι συνοψίζουν σαφώς τα δεδομένα με διαγραμματικό ή οπτικό τρόπο.

Οι μονομεταβλητές προσεγγίσεις ερευνούν μία μεταβλητή (στήλη δεδομένων) κάθε φορά, ενώ οι πολυμεταβλητές μέθοδοι ερευνούν δύο ή περισσότερες μεταβλητές ταυτόχρονα. Το πολυμεταβλητό EDA θα είναι συχνά διμεταβλητό (εξετάζοντας ακριβώς δύο παράγοντες), αλλά περιστασιακά μπορεί να περιλαμβάνει τρεις ή περισσότερες μεταβλητές.

Προτού εκτελεστεί ένα EDA πολλαπλών μεταβλητών, είναι σχεδόν πάντα καλή ιδέα να εκτελείται μονομεταβλητή EDA σε καθένα από τα στοιχεία. Πέρα από τις τέσσερις κατηγορίες που σχηματίστηκαν από την προηγούμενη διαταξινόμηση, κάθε κατηγορία EDA δύναται να υποδιαιρείται περαιτέρω με βάση τον ρόλο (αποτέλεσμα ή επεξηγηματικό) και τον τύπο (κατηγορικό ή ποσοτικό) της υπό εξέταση μεταβλητής [18] [4].

Η EDA ταξινομείται σε τέσσερις τύπους:

1. **Μη γραφική μονομεταβλητή** Αυτός είναι ο πιο βασικός τύπος ανάλυσης δεδομένων, με μία μόνο μεταβλητή να μελετάται. Δεν αντιμετωπίζει αιτίες ή σχέσεις επειδή είναι μια ενιαία μεταβλητή. Ο πρωταρχικός στόχος της μονομεταβλητής ανάλυσης είναι ο χαρακτηρισμός των δεδομένων και ο εντοπισμός προτύπων μέσα σε αυτά.

2. **Γραφική μονομεταβλητή** Οι μη γραφικές προσεγγίσεις δεν παρέχουν πλήρη εικόνα των δεδομένων. Ως αποτέλεσμα, απαιτούνται γραφικές προσεγγίσεις. Τα μονομεταβλητά γραφικά που χρησιμοποιούνται συνήθως περιλαμβάνουν:

    - Γραφικά στελέχους και φύλλα, τα οποία εμφανίζουν όλες τις τιμές δεδομένων καθώς και τη μορφή της διανομής.
    - Τα ιστογράμματα είναι ένας τύπος γραφικής παράστασης ράβδων όπου κάθε ράβδος αντικατοπτρίζει τη συχνότητα (μέτρηση) ή την αναλογία (μέτρηση/συνολική μέτρηση) των εμφανίσεων για ένα δεδομένο σύνολο τιμών.
    - Τα διαγράμματα πλαισίου δείχνουν μια επισκόπηση πέντε αριθμών του ελάχιστου, του πρώτου τεταρτημορίου, του μέσου όρου, του τρίτου τεταρτημορίου

και του μέγιστου.

3. Μη γραφική πολυμεταβλητή: Δεδομένα που προέρχονται από περισσότερες από μία μεταβλητές. Οι πολυμεταβλητές μη γραφικές προσεγγίσεις EDA χρησιμοποιούν συχνά διασταυρούμενη πινακοποίηση ή στατιστικά στοιχεία για να καταδείξουν τη σχέση μεταξύ δύο ή περισσότερων μεταβλητών δεδομένων.

4. Πολυμεταβλητά γραφικά: Τα δεδομένα πολλαπλών μεταβλητών εμφανίζουν σχέσεις μεταξύ δύο ή περισσότερων συνόλων δεδομένων χρησιμοποιώντας γραφικά. Ένα ομαδοποιημένο διάγραμμα ράβδων ή γράφημα ράβδων είναι η πιο συχνά χρησιμοποιούμενη απεικόνιση, με κάθε ομάδα να αντιπροσωπεύει ένα επίπεδο μιας από τις μεταβλητές και κάθε γραμμή μέσα σε μια ομάδα να υποδεικνύει τα επίπεδα της άλλης μεταβλητής.

1.7 Τύποι δεδομένων

Οπότε, σε αυτό το σημείο, μπορούμε να αναρωτηθούμε γιατί πρέπει να γνωρίζουμε αυτές τις διάφορες μορφές δεδομένων για να τις αναλύσουμε. Ο λόγος είναι ότι οι στατιστικές διαδικασίες προορίζονται να λειτουργούν με συγκεκριμένους τύπους δεδομένων και όχι με άλλες. Πολλές προσεγγίσεις για την ανάλυση συνεχών δεδομένων μπορεί να μην είναι κατάλληλες για κατηγορικά δεδομένα. Εάν δεν γνωρίζουμε με τι είδους δεδομένα έχουμε να κάνουμε, μπορεί να γίνει εσφαλμένη ανάλυση. Υπάρχουν, γενικώς, πολλές στατιστικές μελέτες που μπορούν να πραγματοποιηθούν. Η γνώση του τύπου των δεδομένων που έχουμε στη διάθεσή μας περιορίζει τις επιλογές μας και μας επιτρέπει να επιλέξουμε την καλύτερη ανάλυση για αυτά τα δεδομένα.

'Εχοντας καλύψει τις διάφορες μορφές δεδομένων, θα περάσουμε ώστε να δούμε τη μονομεταβλητή ανάλυση. Είναι πολύ σημαντικό να γίνει κατανοητός ο τύπος των δεδομένων πριν χρησιμοποιήσουμε οποιαδήποτε μορφή τεχνικών ανάλυσης. Τι είδους χαρακτηριστικά (στήλες) έχουν τα δεδομένα; Είναι οι πληροφορίες αριθμητικές ή κατηγορικές; Η ανάλυση των δεδομένων είναι κρίσιμη για την κατανόησή αυτών. [7] [6].

1.8 Αλγόριθμοι Συσταδοποίησης

Οι εναλλακτικές λύσεις ή τα συμπληρώματα του εργαλείου προβολής σε λανθάνουσες μεταβλητές που αναφέρθηκαν μέχρι τώρα περιλαμβάνουν προσεγγίσεις ανάλυσης συστάδων.

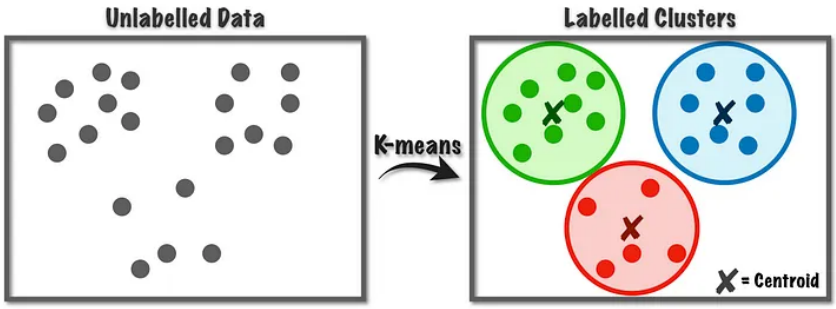
Με βάση την ιδέα ότι τα συγκρίσιμα στοιχεία αντιπροσωπεύονται από κοντινά σημεία στο χώρο των μεταβλητών που τα ορίζουν, ο πρωταρχικός στόχος της ανάλυσης συστάδων είναι να προσδιορίσει ομαδοποιήσεις μέσα σε ένα δεδομένο σύνολο δεδομένων.

Τόσο ο ορισμός των ομάδων όσο και ο αλγόριθμος που χρησιμοποιείται για την κατασκευή τους ποικίλλουν μεταξύ των πιθανών τεχνικών. Ενώ οι αλγόριθμοι ομαδοποίησης βασίζονται σε διάφορους τρόπους για να ορίσουν την εγγύτητα, είτε ως προς την ομοιότητα είτε ως προς την ανομοιότητα, ο ορισμός της ομάδας βασίζεται συνήθως σε μέτρα εντός της ομάδας (για παράδειγμα, υψηλή ομοιότητα μεταξύ των παρατηρήσεων) ή εναλλακτικά σε μέτρα μεταξύ ομάδων (για παράδειγμα, μέγιστη απόσταση μεταξύ των αντικειμένων) [7].

1.8.1 Καθορισμός Βαθμού ομοιότητας

Η πιο φυσική μέθοδος καθορισμού του βαθμού ομοιότητας μεταξύ δειγμάτων βασίζεται στη μετατροπή του πίνακα δεδομένων NxM σε έναν πίνακα NxN αποστάσεων D που λαμβάνεται με τον καθορισμό μιας μετρικής, όπως η Ευκλείδεια απόσταση (η έννοια της ανομοιότητας είναι συμπληρωματική, δηλαδή η τιμή αυξάνεται όσο περισσότερο τα αντικείμενα είναι διαφορετικά, ενώ η ομοιότητα αυξάνεται όσο περισσότερο είναι παρόμοια τα αντικείμενα).

Το εύρος ομοιότητας είναι [0, 1] και όσο πιο κοντά είναι το i και το j το ένα στο άλλο, τόσο μεγαλύτερη είναι η τιμή της ομοιότητας. Είναι αυτονόητο ότι μια ποικιλία μέτρων απόστασης μπορεί να χρησιμοποιηθεί για τη σύγκριση της ομοιότητας διαφόρων αντικειμένων και μια ποικιλία κριτηρίων ομοιότητας μπορούν να οριστούν στον αλγόριθμο. Σε πολλές περιπτώσεις, μπορεί επίσης να είναι ενδιαφέρον να ομαδοποιούνται οι μεταβλητές μαζί αντί για δείγματα. Σε αυτήν την περίπτωση, μία από τις προσεγγίσεις ομαδοποίησης που μπορεί να χρησιμοποιηθεί για αντικείμενα μπορεί να βασίζεται στον συντελεστή συσχέτισης των μεταβλητών ως δείκτης εγγύτητας. Αυτή η τεχνική χρησιμοποιείται συχνά για τη σύνδεση μεταβλητών και θα συζητηθεί εν συντομία σε αυτήν την ενότητα [7] [6].

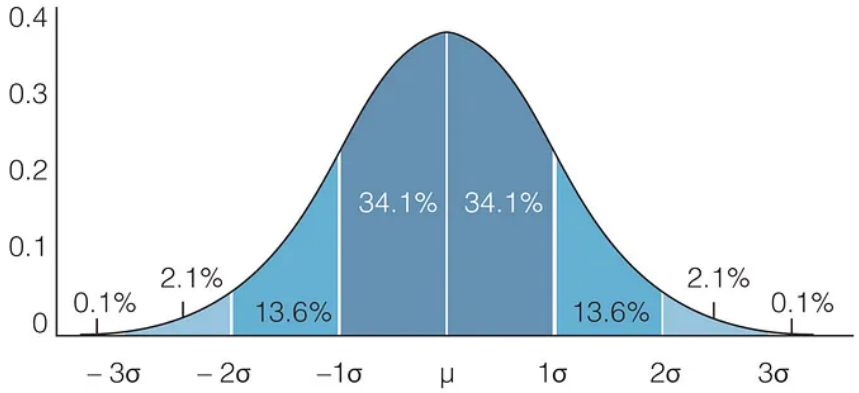


Σχήμα 1.3: Ομαδοποίηση με βάση τον αλγόριθμο k-means, ανάλογα με την ομοιότητα χαρακτηριστικού.

1.8.2 Ομαδοποίηση στοιχείων

Κάθε διαφορετική προσέγγιση ανάλυσης συστάδων εστιάζεται περισσότερο στην ανίχνευση ενός συγκεκριμένου τύπου συμπλέγματος παρά σε άλλους, για παράδειγμα, λειτουργούν καλύτερα όταν τα αντικείμενα σχηματίζουν στρογγυλά, πυκνά συμπλέγματα αντί να έχουν επιμήκεις, επικαλυπτόμενες κατανομές. Αυτή η μεγάλη ποικιλία αλγορίθμων ομαδοποίησης σχετίζεται με το γεγονός ότι τα ίδια τα συμπλέγματα μπορούν να έχουν πολύ διαφορετικά χαρακτηριστικά ως προς το σχήμα, τη διάσταση και την πυκνότητα. Η καλύτερη προσέγγιση είναι στη συνέχεια να αξιολογηθούν τα αποτελέσματα των μεθόδων κατάλληλων για διαφορετικές καταστάσεις και να αντληθούν πληροφορίες για το είδος και τον βαθμό της ομαδοποίησης, που υπάρχει στα δεδομένα, από τα αποτελέσματά τους. Η ομαδοποίηση βάσει μοντέλων είναι δυνατή, αλλά υπερβαίνει τον διερευνητικό σκοπό, επειδή απαιτεί πολλές εκ των προτέρων γνώσεις για το σύστημα [6].

Όπως αναφέρθηκε προηγουμένως σε αυτό το κεφάλαιο, η ομαδοποίηση των στοιχείων μπορεί να διερευνηθεί με διάφορους τρόπους. Ξεκινώντας, παρατηρούμε την κανονική κατανομή. Μπορούμε να δούμε ότι σε μία απόσταση από τη διάμεσο, έχουμε ένα ποσοστό δειγμάτων, το οποίο είναι συμμετρικό. Έτσι σε απόσταση ίση με 1 μονάδα της τυπικής απόκλισης, έχουμε 34.1 τοις εκατό, ενώ σε δύο μονάδες έχουμε επιπλέον 13.6 τοις εκατό. Έτσι, σε τέσσερεις μονάδες τυπικής απόκλισης έχουμε το 95 τοις εκατό των δειγμάτων. Μπορούμε έτσι να εξάγουμε και οπτικά κάποια συμπεράσματα για τη συμπεριφορά στην κατανομή αυτή.



Σχήμα 1.4: Κατανομή της διασποράς πιθανοτήτων σε κανονική κατανομή.

Τόσο οι μέθοδοι προβολής σε λανθάνουσες μεταβλητές όσο και οι μέθοδοι ανάλυσης συστάδων όταν υπάρχουν περισσότερες από τρεις μεταβλητές που πρέπει να ληφθούν υπόψη, μπορεί να έχουν κάποια πληροφορία που αντικατοπρίζεται πάνω από μία φορά. Επιπλέον, οι μέθοδοι προβολής μπορούν να ωφεληθούν από τις μεθόδους ανάλυσης συστάδων, καθώς είναι εφικτή η χρήση μιας προσέγγισης ομαδοποίησης για την εργασία στο νέο σύνολο δεδομένων χαμηλότερων διαστάσεων, όπως οι τιμές Principal component analysis (PCA), κατά την οποία προσπαθούμε να εντοπίσουμε τους άξονες που τα δείγματα κινούνται. Στόχος είναι να βελτιωθεί η ομαδοποίηση των αντικειμένων και να εντοπιστούν ακραίες τιμές. Αυτό είναι ιδιαίτερα χρήσιμο όταν υπάρχει πολύ πληροφορία και πολλές μεταβλητές, γεγονός που καθιστά αναγκαία τη χρήση πολλών διαγραμμάτων διασποράς βαθμολογίας. Κατά την ανάλυση συμπλέγματος μόνο στις βαθμολογίες, τα συμπλέγματα μπορούν να επιθεωρηθούν απευθείας σε ένα μόνο γράφημα, αλλά δεν μπορούν να συλλεχθούν πληροφορίες σχετικά με τους παράγοντες (σε αυτό το παράδειγμα, υπολογιστές) που προκάλεσαν την ανάπτυξη των συστάδων [6] [12].

1.8.3 K-means

Η κατάτμηση και οι ιεραρχικές προσεγγίσεις είναι οι δύο κύριες κατηγορίες που συνθέτουν τους αλγόριθμους ομαδοποίησης. Ο στόχος της κατάτμησης είναι να διαιρεθεί μια μεγάλη συλλογή ετερογενών στοιχείων σε k συστάδες, όπου το k είτε είναι γνωστό εκ των προτέρων, είτε εικάζεται μέσω διερευνητικής ανάλυσης (k-means clustering), είτε επαναληπτικά «βρίσκεται» από τον αλγόριθμο. Αυτή η οικογένεια τεχνικών περιλαμβάνει τα k-means του MacQueen. [10]Από την άλλη πλευρά, η ιεραρχική ομαδοποίηση χρησιμοποιεί μια πολυεπίπεδη αποσύνθεση σε διαφορετικά επίπεδα ομοιότητας - ανομοιότητας και μπορεί να εφαρμοστεί από πάνω προς τα κάτω ή από κάτω προς τα πάνω.

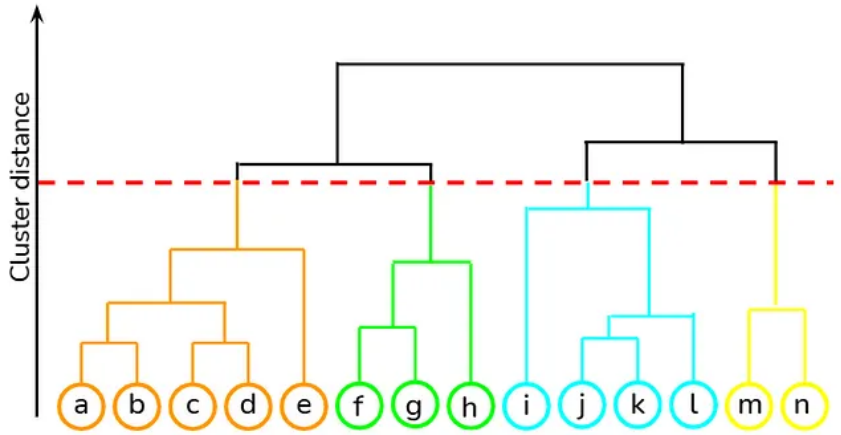
Οι τεχνικές ομαδοποίησης συσσωματώσεων παράγονται χρησιμοποιώντας τη μέθοδο από κάτω προς τα πάνω(bottom-up). Ξεκινούν με κάθε κομμάτι δεδομένων ως ξεχωριστό σύμπλεγμα και προχωρούν ενώνοντας συστάδες με βάση το πόσο όμοια είναι, μέχρι ένα κριτήριο διακοπής. Η απλή σύνδεση, η μέση σύνδεση και η πλήρης σύνδεση είναι μερικά παραδείγματα προσεγγίσεων που αντιπροσωπεύουν αυτήν την οικογένεια. Οι προσεγγίσεις διαιρετικής ομαδοποίησης, οι οποίες ξεκινούν με όλα τα δεδομένα σε ένα ενιαίο σύμπλεγμα και τα χωρίζουν σταδιακά μέχρι να εκπληρωθεί η συνθήκη διακοπής, βασίζονται σε μια μεθοδολογία από πάνω προς τα κάτω.(top-down)

1.8.4 Άλλες τεχνικές

Κάθε τεχνική επιδιώκει να σχηματίσει συστάδες, των οποίων η θέση στον χώρο Μ-διάστασης καθορίζεται από ένα κέντρο, το διάνυσμα των μέσων των μεταβλητών που υπολογίζονται στα συστατικά μέρη του συμπλέγματος. Το δενδρογράφημα είναι ένα εργαλείο που χρησιμοποιείται για την εμφάνιση των αποτελεσμάτων ομαδοποίησης που εμφανίζει γραφικά τον βαθμό ομοιότητας στον οποίο συνδέονται κάθε στοιχείο και σύμπλεγμα [7] [12].

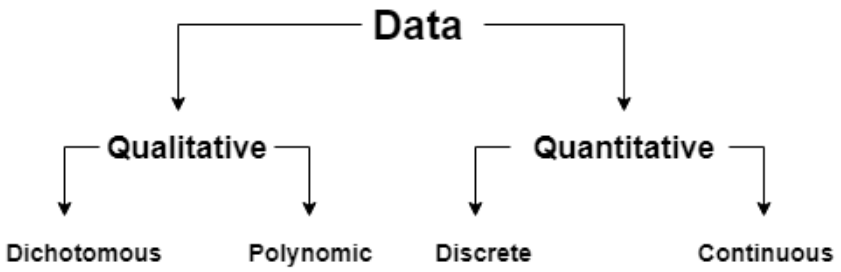
Η προσέγγιση της αθροιστικής ιεραρχικής ανάλυσης συστάδων (HCA) είναι μια από τις πιο εύκολες μεθόδους για να εξηγηθεί πώς λειτουργεί η ανάλυση συστάδων στην πράξη. 1.5 Χρησιμοποιείται μια αναδρομική διαδικασία για να συνεχιστεί η ανάλυση μετά από τα συνηθισμένα προπαρασκευαστικά στάδια που έχουν ήδη αναφερθεί, όπως ο καθορισμός της μετρικής (Ευκλείδειος, Μαχαλανόμπις, απόσταση Μανχάταν, κ.λπ.) και ο υπολογισμός του πίνακα απόστασης και του πίνακα ομοιότητας που ταιριάζει [12]. Τα βήματα έχουν ως εξής:

1. Σημειώνονται τα δύο στοιχεία που είναι πιο συγκρίσιμα
2. Ένα σύμπλεγμα σχηματίζεται από τα δύο στοιχεία που ανακαλύφθηκαν στο σημείο.
3. Υπολογίζεται ο δείκτης ομοιότητας μεταξύ του νέου συμπλέγματος και όλων των άλλων στοιχείων.



Σχήμα 1.5: Ιεραρχική ανάλυση συστάδων (HCA)

Μια νέα σειρά και στήλη που αναφέρουν τον δείκτη ομοιότητας του νέου συμπλέγματος με όλα τα υπόλοιπα αντικείμενα αντικαθιστούν τις σειρές και τις στήλες που σχετίζονται με τα δύο αντικείμενα που μόλις συνδέθηκαν στον πίνακα ομοιότητας, ανεξάρτητα από τη μέθοδο ομαδοποίησης που χρησιμοποιήθηκε για να υπολογίσουμε τον δείκτη ομοιότητας.



Σχήμα 1.6: κατηγορίες ανάλυσης

1.9 Εργαλεία (Software)

Ορισμένα από τα ευρέως γνωστά εργαλεία που χρησιμοποιούνται στην επιστήμη των δεδομένων για τη δημιουργία της EDA είναι[9] [1]:

- **Python:** Είναι μια αντικειμενοστραφής, ερμηνευμένη γλώσσα προγραμματισμού με δυναμική σημασιολογία. Οι υψηλού επιπέδου, ενσωματωμένες δομές δεδομένων της, μαζί με τη δυναμική πληκτρολόγηση και τη δυναμική σύνδεση, το καθιστούν ιδιαίτερα ελκυστικό για γρήγορη δημιουργία εφαρμογών καθώς και χρήση ως γλώσσα σεναρίου για τη σύνδεση υπαρχόντων στοιχείων. Η Python μπορεί να χρησιμοποιηθούν παράλληλα από την EDA για την εύρεση τιμών που λείπουν σε μια συλλογή δεδομένων, κάτι που

είναι χρήσιμο για τον προσδιορισμό του τρόπου χειρισμού των τιμών που λείπουν στη μηχανική εκμάθηση.

- **R:** Η γλώσσα προγραμματισμού ανοιχτού κώδικα του R (Foundation for Statistical Computation) και το περιβάλλον ελεύθερου λογισμικού για στατιστικούς υπολογιστές και γραφικά. Η γλώσσα προγραμματισμού R χρησιμοποιείται συνήθως από τους στατιστικολόγους στην επιστήμη των δεδομένων για τη δημιουργία στατιστικών παρατηρήσεων και την ανάλυση δεδομένων.

- **Matlab:** Το περιβάλλον της Matlab θεωρούνταν μέχρι την εμφάνιση της Python το καλύτερο εργαλείο για επιστημονική χρήση στην επιστημονική κοινότητα. Παρέχει μία μεγάλη ποικιλία πακέτων και βιβλιοθηκών. Η ύπαρξη του περιβάλλοντος κάνει εύκολη την ενσωμάτωση επιπλέον βιβλιοθηκών, ωστόσο η πολυπλοκότητα της γλώσσας, καθώς και η ανάγκη για άδεια για τη λειτουργία του, κάνει τη λύση αυτή όλο και λιγότερο διαδεδομένη.

- **Βιβλιοθήκες και πακέτα:** Διάφορες βιβλιοθήκες δίνουν δυνατότητες σχετικές με την ανάλυση δεδομένων και το EDA. Η πλειονότητα αυτών υποστηρίζει την Python, αλλά δίνεται και η δυνατότητα επιπλέον λύσεων σε αρκετές περιπτώσεις. Ορισμένες από αυτές είναι το Keras, τα Jupyter Notebooks, το Tensorflow, το Apache Spark και το Hadoop. Επιπλεόν, πολλές λύσεις θα κάνουν χρήση και του cloud, με διεπαφές με τους διάφορους providers όπως τα Amazon Sagemake, Azure Machine Learning και το Google Cloud AI, κ.ά.

- **Weka:** Αυτό είναι ένα πακέτο εξόρυξης δεδομένων ανοιχτού κώδικα που περιλαμβάνει πολλά εργαλεία και αλγόριθμους EDA (https://www.cs.waikato.ac.nz/ml/weka/).

- **SPSS IBM:** Είναι μία οικογένεια εφαρμογών για τη χρήση στατιστικών και δεδομένων. Εντός των λύσεων, υπάρχει και ένα εργαλείο οπτικοποίησης και reporting, καθώς επίσης και το SPSS Modeler. Το τελευταίο δίνει ένα γραφικό περιβάλλον για την επιστήμη δεδομένων, και κάνει χρήση των βιβλιοθηκών του SPSS, τα οποία υπάρχουν για χρόνια στην πλατφόρμα της IBM, καθώς η λύση ξεκίνησε να υλοποιείται το 1968.Τέλος, προσφέρεται και ένα περιβάλλον διεπαφής με python και R.

- **KNIME:** Αυτό είναι ένα εργαλείο ανοιχτού κώδικα για ανάλυση δεδομένων και βασίζεται στο Eclipse (https://www.knime.com/).

- **Εργαλεία:** Επιπλέον, υπάρχουν διάφορα εργαλεία που χρησιμοποιούνται από επαγγελματίες και μπορεί να χρησιμοποιούν μία γλώσσα προγραμματισμού, ή να έχουν χαμηλές απαιτήσεις ως προς τον κώδικα που απαιτείται για τη λειτουργία αυτών. Ορισμένα από αυτά τα εργαλεία μπορεί να είναι το PowerBI, το Tableau

Κεφάλαιο 2

Υλοποίηση

2.1 Ξεκινώντας με την EDA

Όπως αναφέρθηκε προηγουμένως, η Python θα χρησιμεύσει ως το κύριο εργαλείο για την ανάλυση δεδομένων. Η Python τοποθετείται συνήθως μεταξύ των 10 κορυφαίων γλωσσών προγραμματισμού και χρησιμοποιείται συνήθως από επαγγελματίες της επιστήμης δεδομένων για ανάλυση δεδομένων και εξόρυξη δεδομένων. Ορισμένα από τα εργαλεία και πακέτα που χρησιμοποιούνται στηνPython είναι τα ακόλουθα:

Προγραμματισμός	**Χαρακτηριστικά**
Python Programming	Βασίζεται σε μεταβλητές, string, τύπους δεδομένων Conditionals Συναρτήσεις Αλληλουχίες, συλλογές, επαναλήψεις Εργασία με αρχεία Αντικειμενοστραφής προγραμματισμός
NumPy	Δημιουργία πινάκων (arrays), αντιγραφή και διαίρεση πινάκων Εκτέλεση διάφορων διεργασιών σε NumPy πίνακες Κατανόηση επιλογής πινάκων Προχωρημένο indexing (ευρετήριο και επέκταση πινάκων Εργασία με πολύ-διάστατους πίνακες Γραμμικές αλγεβρικές συναρτήσεις και ήδη έτοιμες NumPy συναρτήσεις
Pandas	Κατανόηση και δημιουργία DataFrame Δημιουργία ευρετηρίου δεδομένων και υποκατηγοριοποίηση δεδομένων Αριθμητικές συναρτήσεις και χαρτογράφηση Οργάνωση ευρετηρίου Δημιουργία στυλ για οπτική ανάλυση
Matplotlib	Φόρτωση γραμμικών datasets Παραμετροποίηση αξόνων, ετικετών, τίτλων και υπομνημάτων Προσαρμογή αξόνων, κελιών, τίτλων, ετικετών και legends Αποθήκευση γραφημάτων
SciPy	Εισαγωγή του πακέτου Χρήση στατιστικών πακέτων Περιγραφική στατιστική Συμπεράσματα και ανάλυση δεδομένων

Πίνακας 2.1: Χαρακτηριστικά βιβλιοθηκών [11]

2.2 Βάση δεδομένων (περιγραφή του dataset, στοιχεία κλπ)

Αυτή η μέθοδος περιλαμβάνει την πλήρη κατανόηση του συνόλου δεδομένων, των πηγών από τις οποίες συλλέγονται τα δεδομένα, του μεγέθους του συνόλου δεδομένων και των ιδιαιτεροτήτων για κάθε γραμμή και στήλη περιγράφονται σε αυτό το μέρος, το οποίο βοηθά

στην αποτελεσματική ολοκλήρωση του έργου.

Η διαδικασία κατανόησης δεδομένων ασχολείται πρωτίστως με τον προσδιορισμό της ποιότητας των δεδομένων. Η επαλήθευση της ποιότητας των δεδομένων συνεπάγεται τη διασφάλιση ότι τα δεδομένα που επιλέχθηκαν για αυτό το έργο είναι καθαρά και χωρίς σφάλματα και ότι προέρχονται από αξιόπιστη πηγή. Το σύνολο δεδομένων γι' αυτό το έργο ελήφθη από το δημόσιο αποθετήριο δεδομένων Kaggle.

2.2.1 Data Set

Τα στατιστικά στοιχεία αντιστοίχισης περιέχονται στο αρχείο που χρησιμοποιείται γι' αυτό το έργο, το οποίο περιλαμβάνει χιλιάδες σειρές και 49 στήλες.

Πιο συγκεκριμένα, έγινε χρήση του dataset των ATP Tennis Matches από το Kaggle[14] όπου εμπεριέχονται πάνω από 50.000 αγώνες τένις, οι οποίοι παίχτηκαν στο ATP Tour μεταξύ 2000 και 2019. Το σύνολο αυτό, μπορεί να χρησιμοποιηθεί για την ανάλυση διάφορων patterns και με τη χρήση πληροφοριών όπως τα σκορ των αγώνων, οι αγώνες ενός τουρνουά, καθώς και οι πληροφορίες παικτών. Με την ανάλυση του dataset αυτού, είναι δυνατό να γίνει κατανοητό ποια ποσοστά νίκης αντιστοιχούν σε κάθε παίκτη, ενώ μπορεί να γίνει και συσχέτιση μεταξύ παικτών και επιφανειών, ώστε να εξαχθεί μία λογική που μπορεί να αναδείξει πληροφορίες που δεν είναι ορατές με γυμνό μάτι. Με τη χρήση λοιπόν εργαλείων προγραμματισμού, μπορούμε να κάνουμε χρήση του Dataset και να κάνουμε ανάλυση δεδομένων (EDA).

Ορισμένες από τις στήλες που περιλαμβάνονται είναι:

- **Πληροφορίες αγώνων:** Για παράδειγμα, παρέχονται πληροφορίες όπως το ανγνωριστικό αγώνα, καθώς και το όνομα του τουρνουά. Άλλες πληροφορίες είναι ο γύρος, ο τύπος επιφάνείας, η διάρκεια καθώς και το σκορ του αγώνα.

- **Πληροφορίες παίκτη:** Για παράδειγμα, παρέχεται το αναγνωριστικό του παίκτη, οι παίκτες κάθε αγώνα, οι χώρες προέλευσης αυτών καθώς και η κατάταξη τους κατά τον αγώνα

- **Στατιστικά αγώνα:** Η μεγαλύτερη κατηγορία πληροφοριών, που μπορεί να περιέχει τις βαθμολογίες μεμονωμένων σετ, τους κερδισμένους και αποθηκευμένους πόντους μπρέικ, τους κερδισμένους πόντους σερβίς, τους πόντους τκαι τη διάρκεια του κάθε σερβίς, τους άσσους, τα διπλά λάθη, τους νικητές και χαμένους του αγώνα και, τέλος, τον μέσο όρο και τις μέγιστες πιθανότητες.

- **Πληροφορίες αγώνα:** Πληροφορίες που δεν έχουν να κάνουν με το μέρος του παιχνιδιού και μπορεί να είναι για παράδειγμα οι θεατές ενός αγώνα.

Έτσι, με τη χρήση των πληροφοριών που εμπεριέχονται, μπορούμε να αναπτύξουμε στρατηγικές αντιμετώπισης αντιπάλων, προπόνησης, κ.ά. κάνοντας το Data Set μία χρήση πηγή για EDA στον τομέα του τέννις.

2.3 Ανάλυση και Στόχοι (ερωτήματα)

Η ανάλυση που πραγματοποιήθηκε στα δεδομένα είχε σκοπό να απαντήσει σε ενδιαφέροντα ερωτήματα όπως:

- Συνολικές νίκες/ήττες παίκτη
- Ποιος παίκτης έχει περισσότερες νίκες
- Γράφημα νίκης/ήττας ανά έτος
- Κατακτήσεις GrandSlam ανά έτος
- Ποσοστό Άσσων και Διπλών σφαλμάτων
- Ποσοστό πρώτου Σερβίς μέσα
- Πόντοι break που σώθηκαν
- Νίκες/ήττες με αντιπάλους
- Νίκες με βάση την επιφάνεια
- Δυσκολότεροι αντίπαλοι παίκτη
- Συνολική επίδοση ενός παίκτη

Οι επιλογές και τα αποτελέσματα που μπορούμε να αντλήσουμε περιορίζονται από τα σύνολα δεδομένων που μας παρέχονται και από τον κώδικα που θα εκτελέσει την ανάλυση προκειμένου να παράγει σωστά αποτελέσματα. Σε αυτή τη πτυχιακή θα προσπαθήσουμε να απαντήσουμε στα παραπάνω ερωτήματα καθώς και να τα επεκτείνουμε ώστε να αντλήσουμε απαντήσεις, συνδυαστικά.

2.4 Οπτικοποίηση

Η οπτικοποίηση είναι αναπόσπαστο μέρος της EDA, των παρουσιάσεων και των εφαρμογών. Κατά τη διάρκεια της EDA, συχνά εργαζόμαστε μόνοι ή σε μικρές ομάδες και πρέπει να δημιουργήσουμε γραφήματα γρήγορα για να κατανοηθούν καλύτερα τα δεδομένα. Μπορεί να βοηθήσει να εντοπιστούν ανωμαλίες και δεδομένα που λείπουν, καθώς και να προκύψουν άλλα ερωτήματα που οδηγούν σε περισσότερη έρευνα και οπτικοποιήσεις. Συνήθως, αυτό το στυλ αναπαράστασης δεν δημιουργείται με γνώμονα τον τελικό χρήστη. Ο μοναδικός σκοπός του είναι να ενισχύσει την παρούσα κατανόηση. Τα σχέδια δεν χρειάζεται να είναι άψογα.

Κατά την παραγωγή οπτικοποιήσεων για μια εφαρμογή, χρειάζεται να χρησιμοποιήσουμε μια διαφορετική προσέγγιση και να προσέχουμε πολύ τα μικρά πράγματα καθώς κάθε μεταβλητή μπορεί να έχει σημασία. Επιπλέον, συνήθως θα πρέπει να περιορίσουμε τον αριθμό των διαφορετικών αναπαραστάσεων σε ένα αριθμό που αντικατοπτρίζει καλύτερα τα δεδομένα μας. Οι καλές οπτικοποιήσεις δεδομένων κάνουν την εμπειρία εξαγωγής πληροφοριών ευχάριστη για τον θεατή. Οι καλές οπτικοποιήσεις θα περιέχουν πολλές πληροφορίες που κεντρίζουν την προσοχή του θεατή, όπως και οι ταινίες που αιχμαλωτίζουν το κοινό.

Η οπτικοποίηση είναι από τα κύρια εργαλεία του EDA και η data-driven προσέγγιση μπορεί να βοηθήσει στην αποκάλυψη insights καθώς και στην λήψη αποφάσεων. Οι πληροφορίες που θα βασιστούν αυτές οι πληροφορίες θα προέρχονται από τα δεδομένα, μέσα από μία διαδικασία που θα δώσει μία αίσθηση των ιδιοτήτων και των χαρακτηριστικών τους, βοηθώντας στην κατανόηση των χαρακτηριστικών του Dataset και στην ανάλυση αυτού.

Ορισμένες τεχνικές απεικόνισης που χρησιμοποιούνται συνήθως στην EDA περιλαμβάνουν:

- **Ιστογράμματα:** Χρήση για μία μεταβλητή συνήθως, κάνει πιο εύκολη την ανάλυση και τον εντοπισμό ανωμαλιών και ακραίων μοτίβων, παρέχοντας πληροφορίες σχετικά με τη συχνότητα και τη διανομή των δεδομένων.

- **Διαγράμματα διασποράς:** Η σχέση μεταξύ δύο μεταβλητών γίνεται με τη χρήση των διαγραμμάτων διασποράς (scatter plots). Μπορεί να χρησιμοποιηθούν ώστε να αποκαληφθούν συχετίσεις και μοτίβα στα δεδομένα με τη χρήση και των δύο αξόνων.

- **Box plots:** Η κατανομή μίας μεταβλητής γίνεται με τη χρήση box plots, και παρέχονται πληροφορίες σε σχέση με τη διάμεσο, το εύρος και τη μεταβλητότητα των δεδομένων.

- **Χάρτες θερμότητας:** Χρήσιμοι στον εντοπισμό μοτίβων και τις συσχετίσεις μεταξύ μεταβλητών είναι οι χάρτες θερμότητας, οι οποίοι μπορούν να χρησιμοποιηθούν για τη συσχέτιση δύο η περισσότερων μεταβλητών.

- **Συνοπτικές στατιστικές:** Παρέχεται μία περίληψη του συνόλου των δεδομένων με ένα τρόπο συνοπτικό, της κατανομής και της μεταβλητότητας. Γίνεται χρήση επίσης και στατιστικών όπως ο μέσος, η διάμεσος, η τυπική απόκλιση και το εύρος τιμών.

Επιπλέον, στις διαδικασίες το EDA περιλαμβάνονται ο καθαρισμός και το preprocessing των δεδομένων. Η αφαίρεση ή και διόρθωση τυχόν σφαλμάτων, τιμών που λείπουν ή ακραίων τιμών στο σύνολο των δεδομένων είναι διαδικασίες που περιλαμβάνονται στον καθαρισμό των δεδομένων. Στη συνέχεια, μέσω του preprocessing των δεδομένων, μπορεί να γίνει μετατροπή αυτών σεμία μορφή βολικότερη για ανάλυση, με διαδικασίες όπως η επιλογή χαρακτηριστικών, η κλιμάκωση και η κανονικοποίηση.

Για να υπάρξει λοιπόν EDA, είναι να απαραίτητο να διενεργηθεί ανάλυση δεδομένων, και οπτικοποίηση αυτών, καθώς έτσι οι ιδιότητες και τα χαρακτηριστικά των δεδομένων μπορούν να γίνουν κατανοητά, με μία συνοπτική μορφή. Για την οπτικοποίηση, γίνεται χρήση διάφορων βιβλιοθηκών και πακέτων, τα οποία διευκολύνουν τις διαδικασίες αυτές.

2.4.1 Libraries

Ορισμένες από τις βιβλιοθήκες που χρησιμοποιούνται για την εργασία, οι οποίες έχουν να κάνουν κυρίως με τη δημιουργία των dataframes και τη δημιουργία γραφικών παραστάσεων.

2.4.2 Pandas

Η Pandas είναι μέρος του SciPy και έχει πολύ συμπαγείς και ευέλικτες δομές και ρουτίνες για τη διαχείριση αυτών. Επιπλέον, δίνεται η δυνατότητα οπτικοποίησης των δεδομένων σε ένα επιστημονικό φορμάτ δεδομένων. Φυσικά κυρίως γίνεται χρήση πινάκων για διαχείριση δεδομένων.

Επίσης, γίνεται χρήση των σειρών, οι οποίες είναι ένας πίνακας μίας διάστασης με διάφορες ετικέτες και μπορεί να κρατάει δεδομένα κάθε τύπου. Οι ετικέτες οργανώνονται και δημιουργούν συχνά ευρετήρια (index). Μπορούμε να δούμε έτσι και να χειριστούμε δεδομένα στα dataframes, σε μία τετραγωνική δομή δεδομένων.

Ένα dataframe είναι μία δισδιάστατη δομή δεδομένων με στήλες που μπορεί να είναι διάφορων τύπων. Αυτές μπορεί να είναι σειρές, πίνακες διάφορων διαστάσεων, λίστες και λεξικά.

Ένα pandas dataframe έχει μεθόδους οπτικοποίησης όπως και η Matplotlib, την οποία και θα αναλύσουμε αμέσως τώρα.

2.4.3 Matplotlib

Η κύρια βιβλιοθήκη οπτικοποίησης δεδομένων στην Python είναι το matplotlib, ένα έργο που ξεκίνησε στις αρχές της δεκαετίας του 2000 και σχεδιάστηκε για να μιμηθεί τα χαρακτηριστικά χαρτογράφησης του Matlab. Το Matplotlib είναι εξαιρετικά ικανό να χαρτογραφήσει σχεδόν οτιδήποτε μπορεί να φανταστεί κανείς και επιτρέπει στους χρήστες του τεράστιο έλεγχο σε κάθε πτυχή της επιφάνειας σχεδίασης. Ωστόσο, δεν είναι η πιο εύκολη βιβλιοθήκη για να κατανοήσουν οι αρχάριοι.

Η βιβλιοθήκη pandas απλοποιοεί την οπτικοποίηση δεδομένων και συνήθως σχεδιάζει αυτό που θέλουμε με μία μόνο κλήση για γραφική παράσταση. Μία κλήση, εσωτερικά των ρουτίνων της matplotlib θα δημιουργήσει τα διαγράμματα.

2.4.4 Seaborn

Το Seaborn είναι ένα πακέτο οπτικοποίησης που ενσωματώνει το matplotlib και δεν κάνει τη δική του χαρτογράφηση. Το Seaborn δημιουργεί όμορφα γραφήματα και προσφέρει αρκετούς τύπους γραφημάτων σε ποικιλία που δεν είναι διαθέσιμη στην matplotlib ή την pandas. Το Seaborn λειτουργεί με καθαρά (μεγάλα) δεδομένα, αλλά τα pandas υπερτερούν στη συγκέντρωση δεδομένων. Οι μέθοδοι χαρτογράφησης του Seaborn υποστηρίζουν και αντικείμενα DataFrame των pandas.

Παρόλο που είναι δυνατή η δημιουργία γραφημάτων χωρίς να εκτελεστεί ποτέ κώδικας matplotlib, περιστασιακά θα χρειαστεί να τον χρησιμοποιήσουμε για να ρυθμίσουμε χειροκίνητα τις λεπτομέρειες της γραφικής παράστασης.

Εκτός από ένα μικρό αριθμό περιπτώσεων, όλα τα παραδείγματα σχεδίασης θα χρησιμοποιούν pandas ή seaborn.

Η οπτικοποίηση στην Python δεν εξαρτάται μόνο από τη βιβλιοθήκη matplotlib. Υπάρχουν αρκετές βιβλιοθήκες χαρτογράφησης και οι μελλοντικές εκδόσεις των pandas πιθανότατα θα μπορούν να χρησιμοποιούν engines διαφορετικά από το matplotlib. Στον συγκεκριμένο

τομέα υπάρχει ιδιαίτερη δραστηριότητα και οι επιλογές είναι ιδιαιτέρως ενδιαφέρουσες, και προσφέρουν ιδιαίτερα ενδιαφέρουσες επιλογές.

Κεφάλαιο 3

Κώδικας

3.1 Κώδικας

Για το μέρος αυτό, θα αναλύσουμε τον κώδικα, σε επιμέρους τμήματα ανάλογα με τη λειτουργία αυτών.

```
import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
import matplotlib.pyplot as plt
import os
import datetime
import seaborn as sns
```

Αρχίζουμε με την εισαγωγή βιβλιοθηκών απαραίτητων, όπως η numpy, pandas, matplotlib, seaborn για τη σχεδίαση και os, datetime για λειτουργίες συστήματος καθώς και timestamping.

```
def get_aces_percent(row, player_name):
    if row['winner_name'] == player_name:
        val = row.w_ace/row.w_svpt
    elif row['loser_name'] == player_name:
        val = row.l_ace/row.l_svpt
    return val
```

Συνάρτηση για να εξαχθούν οι άσοι ανάλογα με τον κάθε παίχτη σε ποσοστό επιτυχίας

```
def get_double_faults_percent(row, player_name):
    if row['winner_name'] == player_name:
```

```
        val = row.w_df/row.w_svpt
    elif row['loser_name'] == player_name:
        val = row.l_df/row.l_svpt
    return val
```

Συνάρτηση για εξαγωγή των διπλών φάουλ, για κάθε παίχτη, με εξαγωγή σε ποσοστό επιτυχίας.

```
def get_1st_serve_in_percent(row, player_name):
    if row['winner_name'] == player_name:
        val = row.w_1stIn/row.w_svpt
    elif row['loser_name'] == player_name:
        val = row.l_1stIn/row.l_svpt
    return val
```

Συνάρτηση για εξαγωγή των πρώτων σερβίς για κάθε παίχτη, με εξαγωγή σε ποσοστό επιτυχίας.

```
def get_first_serve_win_percent(row, player_name):
    if row['winner_name'] == player_name:
        val = row.w_1stWon/row.w_svpt
    elif row['loser_name'] == player_name:
        val = row.l_1stWon/row.l_svpt
    return val
```

Αντίστοιχα, συνάρτηση για εξαγωγή των δεύτερων σερβίς για κάθε παίχτη και εξαγωγή σε ποσοστό επιτυχίας επίσης.

```
def get_second_serve_percent(row, player_name):
    if row['winner_name'] == player_name:
        val = row.w_2ndWon/row.w_svpt
    elif row['loser_name'] == player_name:
        val = row.l_2ndWon/row.l_svpt
    return val
```

Συνάρτηση για εξαγωγή σε ποσοστό των πόντων που έρχονται μετά από break, σε ποσοστό επιτυχίας. [1]

```
def get_breakpoint_saved_percent(row, player_name):
    if row['winner_name'] == player_name:
        if row.w_bpFaced != 0:
            val = row.w_bpSaved/row.w_bpFaced
        else:
            val = 0
    elif row['loser_name'] == player_name:
        if row.l_bpFaced != 0:
            val = row.l_bpSaved/row.l_bpFaced
        else:
            val = 0
    return val
```

Συνάρτηση για εξαγωγή της αποδοτικότητας ενός παίχτη. ως αποδοτικότητα ορίζεται το ποσοστό των νικών ως προς τους αγώνες(για παράδειγμα σε μία επιφάνεια).

```
# Function to plot effectiveness of a player
def plot_surface_effectiveness(df, player, playerwin):


    pw = df[(df['winner_name'] == player)].groupby(['tourney_year','surface'],
    as_index=False).agg(['count'])
    pww = pw['tourney_id'].reset_index()
    pl = df[(df['loser_name'] == player)].groupby(['tourney_year','surface'],
    as_index=False).agg(['count'])
    pll = pl['tourney_id'].reset_index()
    pww.columns = ['tourney_year','surface','wins']
    pll.columns = ['tourney_year','surface','loses']


    dfs = (pww,pll)
```

[1] Οι πόντοι αυτοί αντιστοιχούν στους πόντους που ένας παίκτης 'σπάει' το σερβίς του άλλου, κερδίζοντας το γκέιμ αυτό.

```
    dfs_concat = pd.concat(dfs, sort=False)
    dfs_final = dfs_concat.fillna(0).groupby(['tourney_year','surface'])
    .agg({'wins':'sum','loses':'sum'}).reset_index()
    dfs_final['r_eff'] = np.where(dfs_final['loses'] > 0,
    dfs_final['wins'] / (dfs_final['wins'] + dfs_final['loses']), 1)
    dfs_final['tourney_year'] = dfs_final['tourney_year'].astype(int)

    g = sns.lmplot(x='tourney_year', y='r_eff', hue='surface', fit_reg=True,
    data=dfs_final, palette='viridis', hue_order=['Hard','Grass','Clay'])
    g.fig.suptitle(player + ' - Effectiveness')
    g.set(xlabel='Year', ylabel='Effectiveness')
    g.set(ylim=(-0.1,1.2))
```

Η αποδοτικότητα θα εμφανιστεί σε διφορετικές επιφάνειες όπως πλαστικό (hard) η οποία είναι και η συνιθέστερη επιφάνεια για τα court, ενώ επίσης υπάρχει το γρασίδι (grass), καθώς και το χώμα(clay).

```
cols = [
    'tourney_id', # Id of Tournament
    'tourney_name', # Name of the Tournament
    'surface', # Surface of the Court (Hard, Clay, Grass)
    'draw_size', # Number of people in the tournament
    'tourney_level', # Level of the tournament
    (A=ATP Tour, D=Davis Cup, G=Grand Slam, M=Masters)
    'tourney_date', # Start date of tournament
    'match_num', # Match number
    'winner_id', # Id of winner
    'winner_seed', # Seed of winner
    'winner_entry', # How the winner entered the tournament
    'winner_name', # Name of winner
    'winner_hand', # Dominant hand of winner (L=Left, R=Right, U=Unknown?)
    'winner_ht', # Height in cm of winner
    'winner_ioc', # Country of winner
```

```
'winner_age', # Age of winner
'winner_rank', # Rank of winner
'winner_rank_points', # Rank points of winner
'loser_id',
'loser_seed',
'loser_entry',
'loser_name',
'loser_hand',
'loser_ht',
'loser_ioc',
'loser_age',
'loser_rank',
'loser_rank_points',
'score', # Score
'best_of', # Best of X number of sets
'round', # Round
'minutes', # Match length in minutes
'w_ace', # Number of aces for winner
'w_df', # Number of double faults for winner
'w_svpt', # Number of service points played by winner
'w_1stIn', # Number of first serves in for winner
'w_1stWon', # Number of first serve points won for winner
'w_2ndWon', # Number of second serve points won for winner
'w_SvGms', # Number of service games played by winner
'w_bpSaved', # Number of break points saved by winner
'w_bpFaced', # Number of break points faced by winner
'l_ace',
'l_df',
'l_svpt',
'l_1stIn',
'l_1stWon',
'l_2ndWon',
```

```
    'l_SvGms',
    'l_bpSaved',
    'l_bpFaced'
]
```

οι στήλες του dataset, μία προς μία, οργανώνονται με την εντολή της pandas.

```
df = pd.concat([
    pd.read_csv('./archive/atp_matches_2000.csv', usecols=cols),
    pd.read_csv('./archive/atp_matches_2001.csv', usecols=cols),
    pd.read_csv('./archive/atp_matches_2002.csv', usecols=cols),
    pd.read_csv('./archive/atp_matches_2003.csv', usecols=cols),
    pd.read_csv('./archive/atp_matches_2004.csv', usecols=cols),
    pd.read_csv('./archive/atp_matches_2005.csv', usecols=cols),
    pd.read_csv('./archive/atp_matches_2006.csv', usecols=cols),
    pd.read_csv('./archive/atp_matches_2007.csv', usecols=cols),
    pd.read_csv('./archive/atp_matches_2008.csv', usecols=cols),
    pd.read_csv('./archive/atp_matches_2009.csv', usecols=cols),
    pd.read_csv('./archive/atp_matches_2010.csv', usecols=cols),
    pd.read_csv('./archive/atp_matches_2011.csv', usecols=cols),
    pd.read_csv('./archive/atp_matches_2012.csv', usecols=cols),
    pd.read_csv('./archive/atp_matches_2013.csv', usecols=cols),
    pd.read_csv('./archive/atp_matches_2014.csv', usecols=cols),
    pd.read_csv('./archive/atp_matches_2015.csv', usecols=cols),
    pd.read_csv('./archive/atp_matches_2016.csv', usecols=cols),
    pd.read_csv('./archive/atp_matches_2017.csv', usecols=cols),
],ignore_index=True) #have to make sure that the index will not be duplicated
```

Στο σημείο αυτό κάνουμε συγκόληση μεταξύ των διάφορων ετών ώστε να έχουμε ένα dataframe το οποίο είναι ενιαίο για όλα τα έτη. Έτσι μπορούμε παράλληλα να έχουμε πρόσβαση και στο έτος, με την προσθήκη της στήλης έτους. Για τις στήλες χρησιμοποιούμε το μοντέλο που δημιουργήσαμε προηγουμένως, με τη βοήθεια της pandas.

Οργανώνουμε τις σειρές με βάση το έτος του τουρνουά και την επιφάνεια.

Οι στήλες, αναλύονται και είναι για παράδειγμα σχετικές με το τουρνουά (ID, όνομα), με την επιφάνεια, το επίπεδο του τουρνουά, καθώς και με τον νικητή και τον ηττημένο του νικητή.

Κάποιες από τις στήλες οι οποίες έχουν ενδιαφέρον για advanced statistics με τον συνδυασμό πληροφορίας, είναι για παράδειγμα το hand of winner, το rank points που ο αθλητής έχει, καθώς και οι seed points.

```
# We add a column with only the year because we will need it in future tasks
df.loc[:,'tourney_date1'] = pd.to_datetime(df['tourney_date'], format='%Y%m%d')
df['tourney_year'] = pd.DatetimeIndex(df['tourney_date1']).year
print(df.head())


winners = list(df.winner_name.unique())
losers = list(df.loser_name.unique())
all_players = list(df.winner_name.unique())
all_players.extend(list(df.loser_name.unique()))
all_players = np.unique(all_players)
```

Η συνάρτηση αυτή χρησιμοποιείται ώστε να εισαχθεί η ημερομηνία του τουρνουά και γίνεται εκτύπωση ενός μέρος του σετ.

Στη συνέχεια δημιουργούμε μεταβλητές για τους νικητές και τους ηττημένους των σετ, ώστε να έχουμε ένα ευρετήριο.

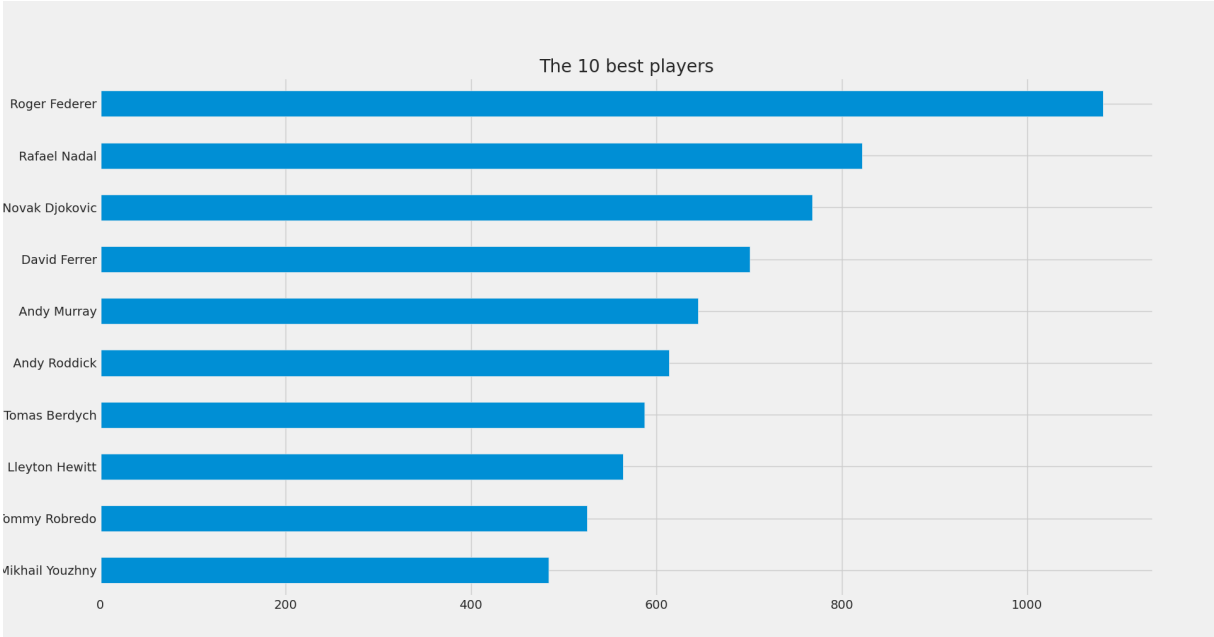
```
                    :~/Desktop/EDA$ sudo python ./etoimo.py
  tourney_id tourney_name surface  ...  l_bpFaced tourney_date1  tourney_year
0   2000-717      Orlando    Clay  ...        4.0    2000-05-01        2000.0
1   2000-717      Orlando    Clay  ...        9.0    2000-05-01        2000.0
2   2000-717      Orlando    Clay  ...       10.0    2000-05-01        2000.0
3   2000-717      Orlando    Clay  ...       11.0    2000-05-01        2000.0
4   2000-717      Orlando    Clay  ...        6.0    2000-05-01        2000.0

[5 rows x 51 columns]
```

Σχήμα 3.1: Data Head

Ξεκινώντας την εκτύπωση των αποτελεσμάτων, πρώτα εμφανίζουμε τους 10 καλύτερους παίκτες με βάση των αριθμό νικών στο dataset.

```
# Get the top 10 players
plt.figure(figsize=(9,5))
```

```
df['winner_name'].value_counts()[:10].sort_values(ascending=True)
.plot(kind='barh')
plt.title('The 10 best players')

top10_players = ["Roger Federer",
"Rafael Nadal",
"Novak Djokovic",
"David Ferrer",
"Andy Murray",
"Andy Roddick",
"Tomas Berdych",
"Lleyton Hewitt",
"Tommy Robredo",
"Mikhail Youzhny"]

years = []
for i in range(0,18):
    years.append(i+2000)
```



Σχήμα 3.2: All players

Με αισθητή διαφορά πρώτος είναι ο Roger Federer, ενώ οι Rafael Nadal και Novak Djokovic

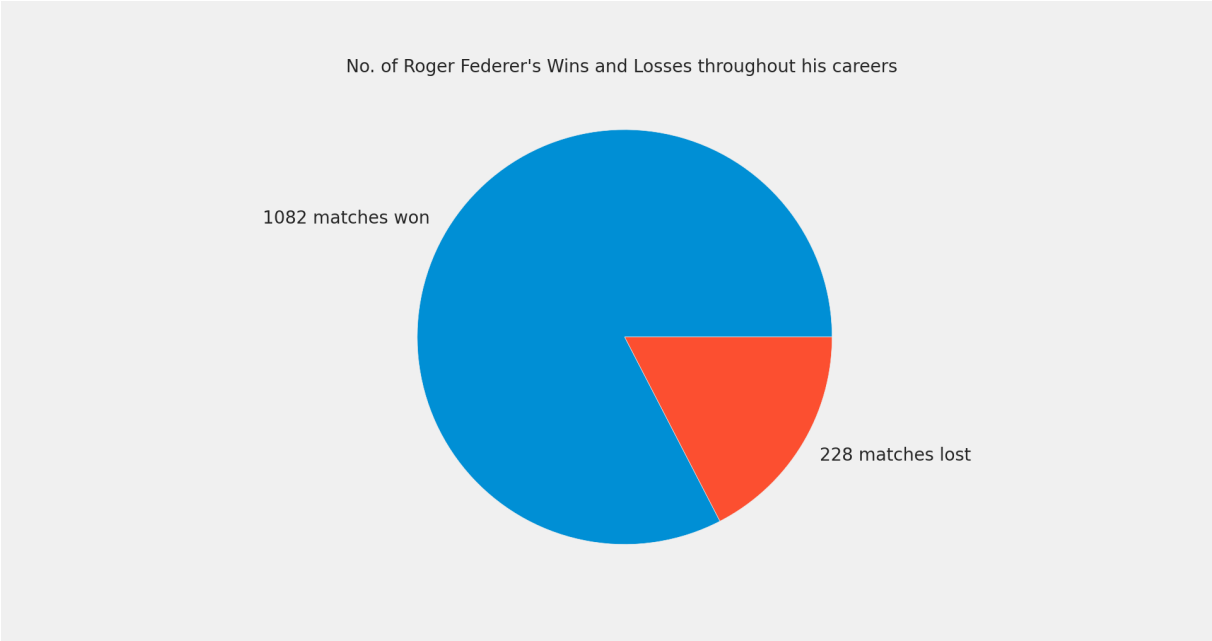
συμπλήρωσαν την τριάδα των καλύτερων παικτών.

```
# 1. Overall Performance of each top player
for player_name in top10_players:
player = df.loc[(df['winner_name'] == player_name) |
(df['loser_name'] == player_name)].copy()
#we want to analyze his performance over time but date is not float 64,
so let's change it to datettime datatype.
player.tourney_date.apply(lambda x: '%.0f' % round(x,0))
player.loc[:,'tourney_date'] =
pd.to_datetime(player['tourney_date'], format='%Y%m%d')
```

Για κάθε παίχτη, παρατηρούμε ποια ήταν η απόδοσή του, και αναλύουμε τον αριθμό νικών και ηττών.

```
# Let's look at his serve performance over the time.
#However, we have to look at the number of serve whether it is winner
#or loser based on loser or winner
# how important is the serve
# let's define variable that we will use a lot
playerwin = player.loc[player['winner_name'] == player_name].copy()
playerloss = player.loc[player['loser_name'] == player_name].copy()
# print(f'Number of wins: {playerwin.count()[0]}')
# print(f'Number of losses: {playerloss.count()[0]}')
#let's plot the number of wins and losses from 2000 to 2017
fig = plt.figure(figsize=(5,5))
plt.title("No. of " + player_name + "'s Wins and Losses throughout his careers ")
plt.pie([playerwin.count()[0],playerloss.count()[0]],
labels = [f'{playerwin.count()[0]} matches won',
f'{playerloss.count()[0]} matches lost'],textprops={'fontsize': 20})
plt.show()
```

Με τη συνάρτηση αυτή δημιουργείται η εικόνα 3.3 κατά την οποία έχουμε σε ένα διάγραμμα πίτας της νίκες και της ήττες κάθε παίκτη. Στη συγκεκριμένη περίπτωση παρατηρούμε ότι ο Roger Federer είχε 1082 νίκες και 228 ήττες.



Σχήμα 3.3: Federer Νίκες και ήττες

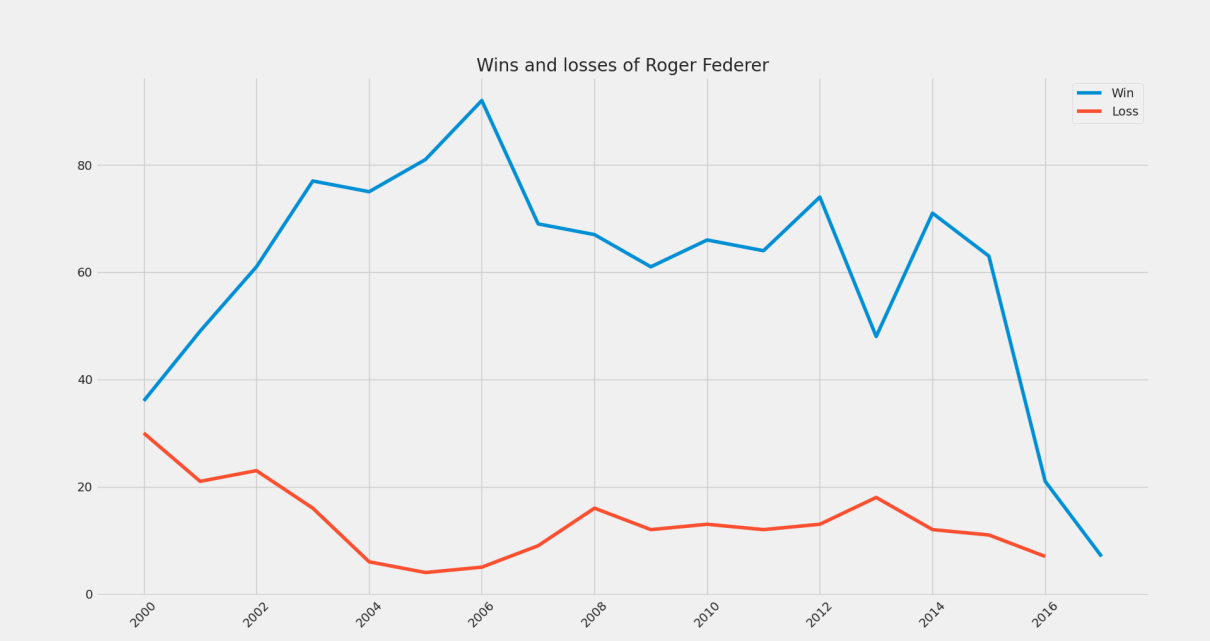
```
# 2. Trend of win and loss in each year
# Get years the plyer is participating
year = playerwin.groupby(playerwin.tourney_date.dt.year).tourney_date.unique()
yls = list(year.index.values)
print(yls)
year = playerloss.groupby(playerloss.tourney_date.dt.year).tourney_date.unique()
yls1 = list(year.index.values)
print(yls1)
annualwin = playerwin.groupby(playerwin.tourney_date.dt.year).
count().tourney_id
annualloss = playerloss.groupby(playerloss.tourney_date.dt.year).
count().tourney_id
print(len(annualwin))
print(len(annualloss))
plt.xticks(np.arange(yls[0], yls[-1], 2), rotation=45)
plt.title("Wins and losses of " + player_name )
plt.plot(yls, annualwin, label='Win')
plt.plot(yls1, annualloss, label='Loss')
plt.legend()
```

```
plt.show()
```

Με τη συνάρτηση αυτή δημιουργούμε ένα γράφημα κατά το οποίο μπορούμε να παρατηρήσουμε την πορεία ενός παίχτη και τις νίκες καθώς και τις ήττες που είχε, σε κάθε έτος, για όλη τη διάρκεια του χρόνου που αναλύουμε στο dataset.

Παρατηρούμε ότι από το 2000 έως και το 2003 ο Roger Federer είχε συνεχώς ανοδική πορεία. Στη συνέχεια αφού είχε μία μικρή (σχετική) κάμψη από τις 80 νίκες, έφτασε περίπου τις 90 νίκες το 2006. Το έτος αυτό φαίνεται να ήταν το καλύτερο για τον Federer, όπου στη συνέχεια διατηρήθηκε έως και το 2012 σε πολύ υψηλά επίπεδα, μεταξύ των 60 και 80 νικών.

Το έτος 2013 φαίνεται να έχει μία κακή πορεία για το επίπεδό του.Μεταξύ 2014 και 2015 φαίνεται να έχει ένα καλό επίπεδο, στα προηγούμενά του στάνταρ. Στη συνέχεια όμως, λόγω και διάφορων τραυματισμών, επήλθε μία πτώση στην πορεία του παίχτη.



Σχήμα 3.4: Federer Νίκες και ήττες σε διάγραμμα ανά τα χρόνια

```
# 3. Number of grandslams
tour = playerwin.loc[playerwin.tourney_level ==
'G'].groupby(playerwin.tourney_id).count()
championship = tour.loc[tour.tourney_id == 7]
plt.title("No. of " + player_name + "'s Grandslam Championships")
plt.yticks([1,2,3])
grandslams = championship.
groupby(championship.index.map(lambda x: x[0:4])).count()
```

```
plt.bar(grandslams.index, grandslams.tourney_id)
plt.show()
```

Στο επόμενο διάγραμμα (3.5) παρατηρούμε τον αριθμό των Grandslam αγώνων που κερδίθηκαν από τον Roger Federer. Για υπενθύμιση, ο τύπος αυτών των αγώνων είναι η υψηλότερη διάκριση του αγωνίσματος.

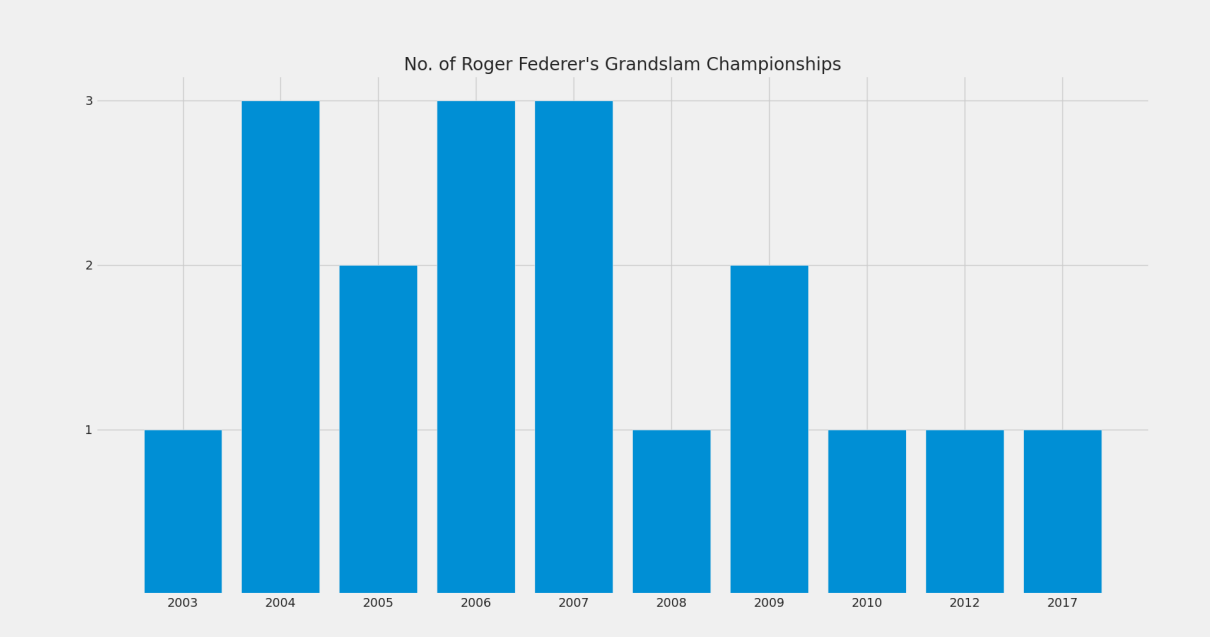
Οι αγώνες αυτοί είναι τέσσερεις και αντιστοιχούν στους εξής:

- Αυσταλιναό Όπεν
- Αμερικανικό Όπεν
- Γουίμπλετον
- Γαλλικό Όπεν (Ρολάν Γκαρός)

Σε συνάρτηση με το προηγούμενο γράφημα, παρατηρούμε ότι μεταξύ 2004 και 2007, ο Roger Federer κατέκτησε την πλειονότητα των τουρνουά που ήταν διαθέσιμα.

Στη συνέχεια, κατέκτησε κάποιον τίτλο, ωστόσο ο αριθμός των τίτλων ήταν μικρότερος από το πρόσφατο παρελθόν.

Μέχρι πρόσφατα, ο Roger Federer κατείχε το ρεκόρ των περισσότερων τίτλων Grandslam με είκοσι τίτλους. Το ρρεκόρ του ξεπεράστηκε από τους επόμενους δύο στη λίστα, τον Rafael Nadal και τον Novak Djokovic. Τη στιγμή που γράφονται αυτές οι γραμμές, και οι δύο παίκτες έχουν εικοσιδύο τίτλους. Για τον Djokovic, αναμένεται αυτός ο αριθμός να αυξηθεί και να έχει το απόλυτο ρεκόρ.



Σχήμα 3.5: Federer Νίκες σε grand slams

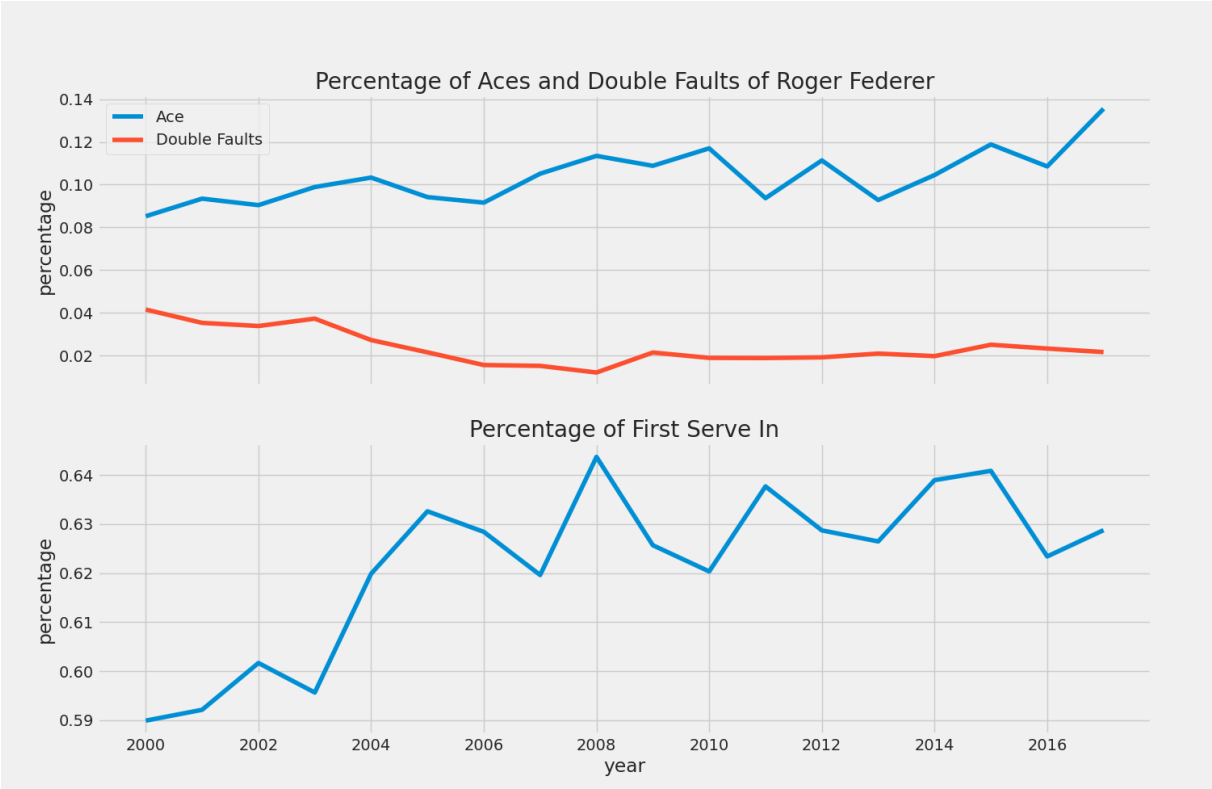
```
# 4. Serve performance by
# a) number of his aces/servepoints, average percentage of ace per match!
# b) number of his double faults/servepoints
# c) 1stin = 1st serve in^^I
player['player_aces_percentage'] =
player.apply(get_aces_percent, args=(player_name,), axis=1)
player['player_double_faults_percentage'] =
player.apply(get_double_faults_percent, args=(player_name,), axis=1)
player['player_1st_serve_in_percent'] =
player.apply(get_1st_serve_in_percent, args=(player_name,), axis=1)
player['player_first_serve_win_percentage'] =
player.apply(get_first_serve_win_percent, args=(player_name,), axis=1)
player['player_second_serve_win_percentage']
 player.apply(get_second_serve_percent, args=(player_name,), axis=1)

#group by year
groupbyyear = player.groupby(player.tourney_date.dt.year).mean()

#fig = plt.figure()
```

```
fig, (ax1,ax2) = plt.subplots(
nrows=2,
ncols=1,
sharex=True
)

fig.set_size_inches(15,10)
fig.subplots_adjust(wspace=0.5)

ax1.set_title('Percentage of Aces and Double Faults of '+ player_name )

ax1.set_ylabel('percentage')
ax1.set_xticks(np.arange(2000,2017,2))
ax1.plot(groupbyyear.index,groupbyyear['player_aces_percentage'], label='Ace')
ax1.plot(groupbyyear.index,groupbyyear['player_double_faults_percentage'],
label='Double Faults')
ax1.legend()

ax2.plot(groupbyyear.index,groupbyyear['player_1st_serve_in_percent'])
ax2.set_ylabel('percentage')
ax2.set_title('Percentage of First Serve In')
ax2.set_xlabel('year')

plt.show()
```

Στο γραφικό 3.6 μπορούμε να παρατηρήσουμε τη γενικότερη τάση και των προηγούμενων γραφημάτων. Οι άσοι αποτελούσαν ένα από τα δυνατά χαρτιά στο παιχνίδι του Federer, και για αυτό και παρατηρούμε μία σχετική σταθερότητα περίπου στο δέκα τοις εκατό, για όλη τη διάρκεια τη περιόδου που αναλύουμε. Παράλληλα με αυτό, μπορούμε να παρατηρήσουμε ότι η τάση αυτή είναι ελαφρά αυξητική ανά τα χρόνια. Από τα πρώτα έτη ανάλυσης με τιμή στο 0.08 τοις εκατό, φτάσουμε στα τελευταία έτη της καριέρας του Federer να έχουμε μία τιμή ίση με 0.16 τοις εκατό των σερβίς του παίχτη. Τη στιγμή λοιπόν που δεν κατέκτησε περισσότερους τίτλους, ο Federer εξέλιξε το παιχνίδι του και κέρδισε περισσότερους πόντους

με άσσους.

Την ίδια στιγμή, όπως παρατηρούμε στην κόκκινη γραμμή του γραφήματος, βλέπουμε ότι τα διπλά λάθη μειώθηκαν κατά τα έτη που ο Federer εξελίχθηκε σε κορυφαίο παίχτη , και κατά τα έτη κορύφωσης της καρίερας του, οι τιμές παρέμειναν στις χαμηλότερες τιμές. Στη συνέχεια του dataset, παρατηρούμε μία ελαφρά αύξηση των λαθών, ωστόσο φαίνεται ότι καθώς με τα χρόνια το παιχνίδι του ωρίμασε, ο αριθμός αυτός δεν ανέβηκε σε πολύ υψηλά νούμερα.



Σχήμα 3.6: Federer Άσοι και πρώτα σερβίς

Παράλληλα, στο δεύτερο γράφημα παρατηρούμε μία απότομη αύξηση στην υποδοχή των πρώτων σερβίς την περίοδο που το παιχνίδι του Federer βελτιώθηκε. Είναι άξιο αναφοράς ότι από το 2004 και μέχρι και το τέλος του dataset, που αντιστοιχεί στο 2017, η ελάχιστη τιμή που παρατηρούμε για τον Federer είναι το 62 τοις εκατό, ενώ σε ορισμένα έτη η τιμή έφτασε και το 65 τοις εκατό. Σε συνδυασμό με τη βελτίωση των άσων στο παιχνίδι του, ο Federer έτσι δημιουργούσε πολλά προβλήματα στους αντιπάλους του και είχε ένα πολύ υψηλό πλεονέκτημα.

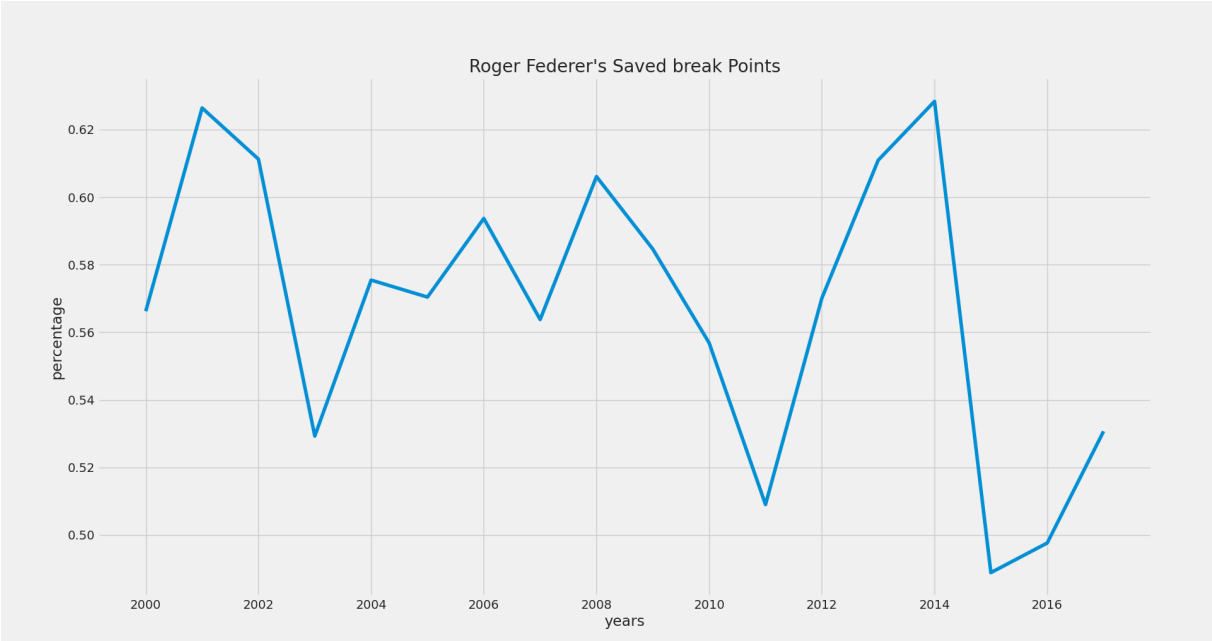
```
# 5. Mentality over the time
player['player_breakpoint_saved_percent'] = player.
```

```
apply(get_breakpoint_saved_percent, args=(player_name,), axis=1)

mental_level = player.groupby(player['tourney_date'].dt.year).mean().
player_breakpoint_saved_percent

plt.title(player_name +"'s Saved break Points")
plt.xlabel('years')
plt.ylabel('percentage')
year = playerwin.groupby(playerwin.tourney_date.dt.year).tourney_date.unique()
yls = list(year.index.values)
plt.xticks(np.arange(yls[0],yls[-1],2))
# plt.plot(np.arange(2000,2018),mental_level)
plt.plot(yls, mental_level)
plt.show()
```

Ένα ακόμη σημαντικό μέρος του παιχνιδιού που είναι πολύ κρίσιμο, είναι οι Break points. [2] Έχοντας ο ίδιος το σερβίς στη συνέχεια, αυτό είναι ένα μέρος του παιχνιδιού που μπορεί να είναι πολύ κρίσιμο, αφού μπορεί να αποβεί καθοριστικό για τη νίκη σε ένα σετ, και ως εκ τούτου, και σε ένα παιχνίδι.

Παρατηρούμε λοιπόν, σε συνάρτηση με τα προηγούμενα γραφήματα ότι ο Roger Federer, είχε ανοδική πορεία και σε αυτή τη μετρική μετά το 2002 όπου και υπήρξε κορυφαίος. Είναι αξιοπρόσεκτο ότι κατά τα πρώτα έτη ανάλυσης έφτασε σε τιμές περίπου ίσες με 65 τοις εκατό, τιμές που δεν επανέλαβε παρά μόνο προς το τέλος της καριέρας του.

[2] Οι πόντοι αυτοί αντιστοιχούν στους πόντους που ένας παίκτης 'σπάει' το σερβίς του άλλου, κερδίζοντας το γκέιμ αυτό.



Σχήμα 3.7: Federer break πόντοι

Παρατηρούμε λοιπόν ότι από το 2003 έως το 2011 είχε αρκετά καλά ποσοστά στη μετρική αυτή. Το 2011 είχε μία πτώση, ανεβαίνοντας για τρία χρόνια συνεχώς, ώσπου και έφτασε σε υψηλό όπως και το 2001. Στη συνέχεια το 2013 φαίνεται να είχε τη χαμηλότερη τιμή, ένδειξη της πτώσης του παιχνιδιού του, ενώ και στη συνέχεια δεν επανήλθε στα υψηλά στάνταρ του παρελθόντος μέχρι και το τέλος των τιμών του dataset.

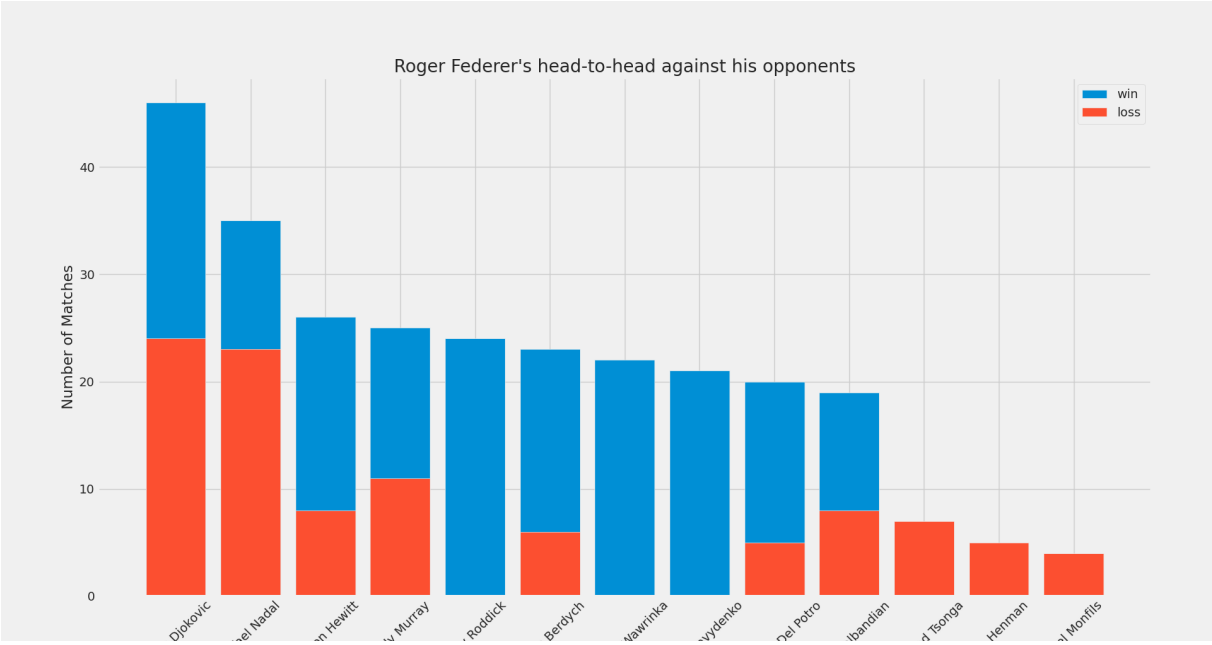
```
# 6. Opponents throughout the year

loss = player.loc[player.winner_name != player_name, ['winner_name']]
# loss.rename(r={'winner_name': 'name'})
loss.columns = ['name']
loss['status'] = 'loss'
win = player.loc[player.winner_name == player_name, ['loser_name']]
win.columns = ['name']
win['status'] = 'win'
opponents = pd.concat([win,loss])
# opponents
#opponents = opponents.groupby('name').count().sort_values('status',
#ascending = False)
```

```
numberofmatches = opponents.groupby('name').count().sort_values('status',
ascending = False)
numberoflosses = opponents.loc[opponents.status == 'loss'[:].
groupby('name').count().sort_values('status', ascending = False)

fig = plt.figure(figsize=(10,5))

plt.xticks(rotation=45)
plt.xlabel('opponents')
plt.ylabel('Number of Matches')

plt.bar(numberofmatches.index[0:10],numberofmatches.status[0:10], label='win')
plt.bar(numberoflosses.index[0:10],numberoflosses.status[0:10], label='loss')
plt.legend()
plt.title( player_name + "'s head-to-head against his opponents")
plt.show()
```

Στο επόμενο γράφημα παρατηρούμε τους αγώνες του Roger Federer με διάφορους αντιπάλους. Στον οριζόντιο άξονα παρατηρούμε σε κάθε ράβδο τον κάθε παίχτη, ενώ στον κάθετο άξονα παρατηρούμε τους αγώνες που δόθηκαν με κάθε παίκτη. Με μπλε απεικονίζονται οι νίκες έναντι αυτού του παίχτη ενώ με πορτοκαλί απεικονίζονται οι ήττες έναντι αυτού του παίχτη.

Τον μεγαλύτερο αριθμό αγώνων έδωσε με τον Novak Djokovic, όπου έδωσε περίπου 50 αγώνες. Τον αμέσως επόμενο αριθμό αγώνων, έδωσε με τον Rafael Nadal, περίπου 35 αγώνες. Έναντι και των δύο οι ήττες ήταν περίπου στο ίδιο επίπεδο (περί τις είκοσι ήττες). Το ποσοστό των ηττών στους αγώνες έναντι του Rafael Nadal ωστόσο, ήταν υψηλότερο.

Αρκετά υψηλό ποσοστό ηττών σε σχέση με τους αγώνες παρατηρούμε ότι υπήρχαν στους αγώνες έναντι του David Nalbandian και του Andy Murray. Ιδιαιτέρως στους αγώνες με τους Tsonga, Henman και Monfils ο Federer δεν είχε καμία νίκη. Στην αντίπερα όχθη, στους αγώνες με τους Rodrick, Wawryncka και Davydencko ο Federer δεν είχε καμία ήττα.

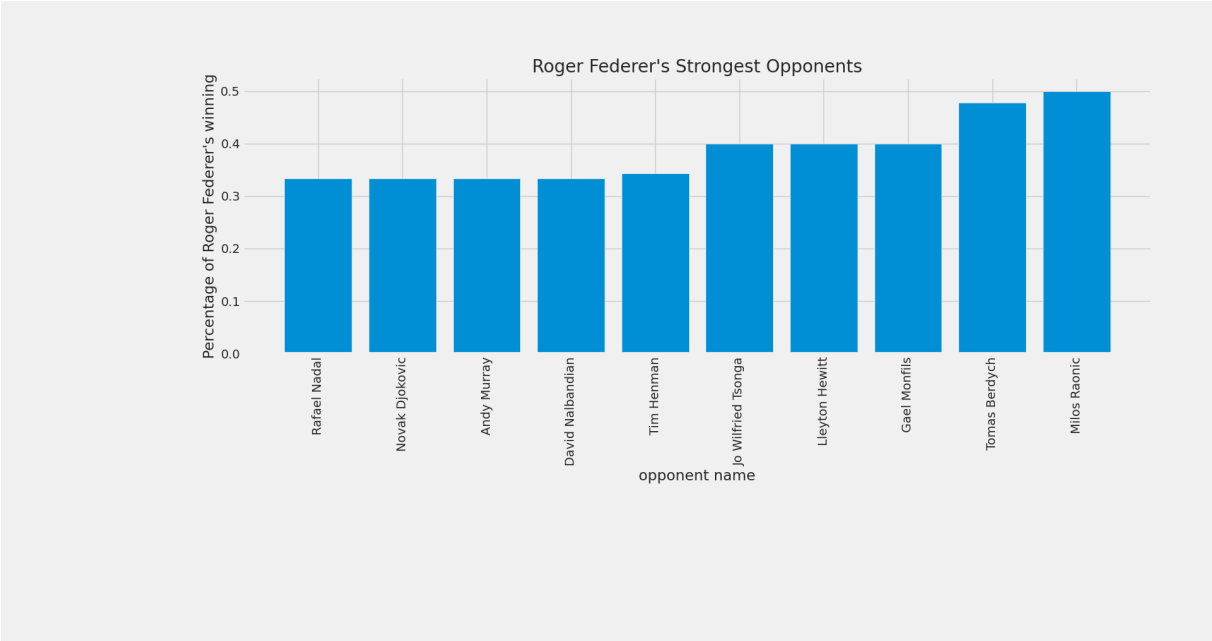


Σχήμα 3.8: Federer head to head

```
# 7. Strongest opponents
opponents_grouped = pd.DataFrame()
opponents_grouped['No. of matches'] = numberofmatches.status
opponents_grouped['No. of winning'] =
opponents.loc[opponents.status=='win'].groupby('name').count().status
opponents_grouped['percentage'] = opponents_grouped['No. of
winning']/opponents_grouped['No. of matches']
opponents_grouped.loc[opponents_grouped['No. of matches'] >
10].sort_values('percentage',ascending = True).head()

opponents.loc[opponents.status=='win'].groupby('name').count()
plt.xlabel('opponent name')
plt.ylabel("Percentage of " + player_name + "'s winning")
plt.xticks(rotation='vertical')
plt.subplots_adjust(bottom=0.45)
plt.subplots_adjust(left=0.2)
# plt.bar(opponents_grouped.sort_values('percentage',ascending = True)
.index[0:10], opponents_grouped.sort_values('percentage',ascending =
True).percentage[0:10])
```

```
plt.bar(opponents_grouped.loc[opponents_grouped['No. of matches'] >
10].sort_values('percentage',ascending = True).index[0:10],
opponents_grouped.sort_values('percentage',ascending = True).percentage[0:10])
plt.title(player_name + "'s Strongest Opponents")
```

Έτσι φτάνουμε στο επόμενο γράφημα, στο οποίο βλέπουμε τους δυνατότερους αντιπάλους του Roger Federer από το Top 10. Στον οριζόντιο άξονα και πάλι παρατηρούμε κάθε παίκτη, ενώ στον κάθετο άξονα το ποσοστό νικών του Roger Federer. Φυσικά, έναντι των δυσκολότεροι αντιπάλων, ο Federer αναμένεται να έχει το χαμηλότερο ποσοστό νικών.



Σχήμα 3.9: Federer αντίπαλοι

Με τους Djokovic, Nadal, Murray και Nalbandian το ποσοστό επικράτησης είναι περίπου στο 35 τοις εκατό. Αντιθέτως, ο Milos Raonic ήταν ο παίχτης από τον οποία ο Federer πήρε τις περισσότερες νίκες.

```
# 8. Plot effectiveness of player based on the surface of the field
plot_surface_effectiveness(df, player_name, playerwin)
```

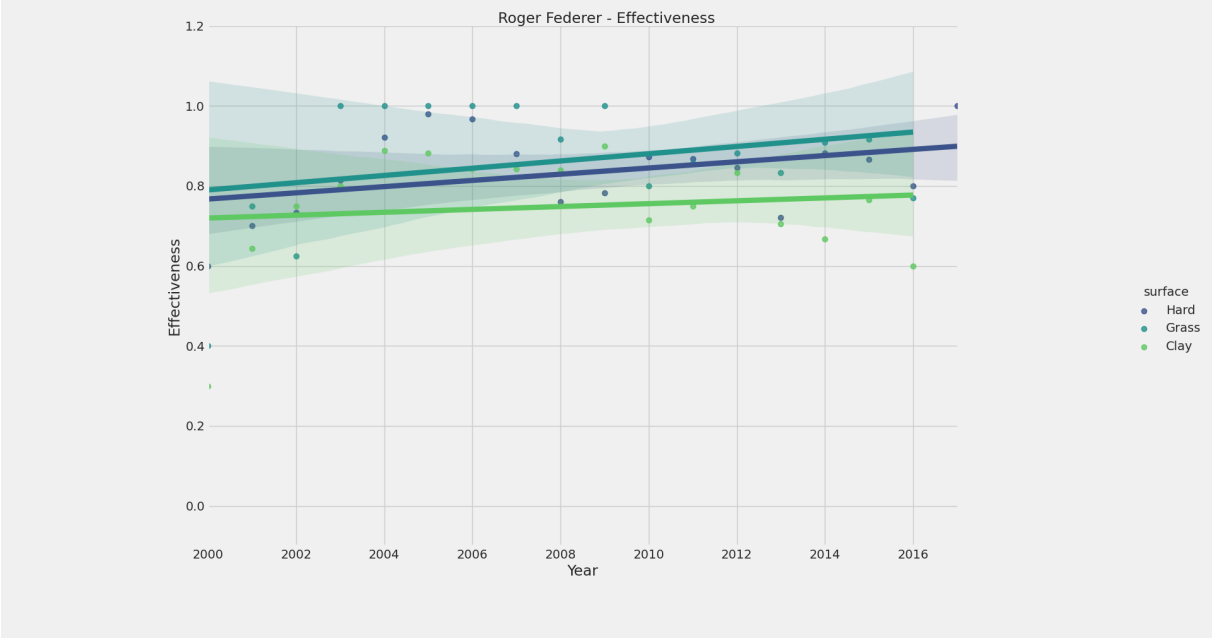
Στο τελευταίο γράφημα παρατηρούμε την αποτελεσματικότητα του Roger Federer. Στον οριζόντιο άξονα έχουμε τα έτη, ενώ στον κάθετο άξονα έχουμε το ποσοστό αποδοτικότητας του Roger Federer στους αγώνες που έδωσε σε κάθε επιφάνεια. Οι τρεις επιφάνειες είναι η

σκληρή επιφάνεια (Hard), το γρασίδι (Grass) και τέλος, το χώμα (Clay). Σε κάθε έτος υπάρχει μία τελεία που αντιστοιχεί στο ποσοστό αποδοτικότητάς του Federer. Στη συνέχεια δημιουργούμε τη βέλτιστη ευθεία που ενώνει αυτά τα σημεία ώστε να αναπαραστήσουμε την ευθεία που έρχεται όσο το δυνατόν πιο κοντά στην πορεία του Federer.

Ιστορικά είναι γνωστό ότι το γρασίδι ήταν η αγαπημένη επιφάνεια του Federer, σε αντίθεση με το χώμα, στο οποίο κυριαρχούσε ο Nadal. Παρατηρούμε δε, ότι μεταξύ 2003 και 2010, ο Federer έχει το απόλυτο στο γρασίδι σχεδόν σε κάθε έτος. Το 2008 όπου δεν έχει το απόλυτο, και πάλι η επιφάνεια αυτή βρίσκεται υψηλότερα από τις άλλες δύο, επιβεβαιώνοντας την κυριαρχία του στη συγκεκριμένη επιφάνεια.

Τέλος, παρατηρώντας τη βελτίωση του Federer ανά τα χρόνια, βλέπουμε ότι η αποδοτικότητα του Federer αυξάνεται σε όλες τις επιφάνειες. Η χαμηλότερη αύξηση φαίνεται να είναι στο χώμα, όπου η γραμμή είναι σχεδόν σταθερή μεταξύ 0.7 και 0.75. Στη συνέχεια βλέπουμε τη σκληρή επιφάνεια, όπου υπάρχει μία αυξητική τάση και από το 0.79 φτάνουμε πάνω από το περίπου στο 0.85 προς το τέλος της καριέρας του.

Ειδική αναφορά πρέπει να γίνει και πάλι στο γρασίδι. Παρατηρούμε ότι η αυξητική τάση είναι μεγαλύτερη εν συγκρίσει με τη σκληρή επιφάνεια, επιβεβαιώνοντας την κυριαρχία του Federer στη συγκεκριμένη επιφάνεια. Η γραμμή της αποδοτικότητας ξεκινάει από το 0.8 και φτάνουμε, προς το τέλος της καριέρας τους Federer να έχουμε μία τιμή περί το 0.9.



Σχήμα 3.10: Federer effective

Κεφάλαιο 4

Συμπεράσματα

4.1 Σχολιασμός

Συνολικά παρατηρούμε ότι, ο Federer ήταν ο καλύτερος παίκτης για τη διάρκεια της ανάλυσης,ενώ πολύ κοντά βρίσκονταν ο Ναδάλ και ο Τζόκοβιτς. Κάτι που μπορούμε να καταλάβουμε, είναι ότι τα grand slam έχουν μεγάλη σημασία στην κατάταξη και τα χρόνια κυριαρχίας του, ο Roger Federer είχε πολύ υψηλό ποσοστό επιτυχίας. Το σερβίς, ήταν πολύ σημαντικό πολύ σημαντικό για το παιχνίδι του, ενώ παρατηρούμε ότι κατά την περίοδο της πτώσης του στην κατάταξη,είχε παράλληλη αύξηση λαθών. Επίσης, με τα χρόνια βελτίωσε το παιχνίδι του και είχε αύξηση στους άσσους. Αγαπημένη επιφάνεια του Federer φαίνεται να είναι το πλαστικό, καθώς στη συγκεκριμένη επιφάνεια είχε πολλοί υψηλά επιτυχίας. Υπήρξαν αθλητές που ήταν αναμενόμενο να ήταν οι δυσκολότεροι του αντίπαλοι (Nadal,Djokovic), ωστόσο μέσω της ανάλυσης είδαμε ότι ανακαλύπτουμε και άλλους αθλητές οι οποίοι δυσκόλεψαν τον Federer.

Είδαμε έτσι λοιπόν, πως μπορεί να γίνει χρήση της EDA ώστε να ανακαλυφθούν πληροφορίες σε ένα μεγάλο χρονικό εύρος, και για ένα μεγάλο αριθμό πληροφοριών. Μέσα από την ανάλυση παρατηρήσαμε τάσεις και ανακαλύψαμε τους καλύτερους αθλητές για τη διάρκεια της ανάλυσης, βλέποντας πληροφορίες για το παιχνίδι τους και τους αγώνες που έπαιξαν.

Υπήρξε μία προσπάθεια να γίνει περαιτέρω ανάλυση για τα πλαίσια της εργασίας, δίχως επιτυχία, ώστε να εξαχθούν πληροφορίες με μεγαλύτερη δυσκολία.

Κάποια από αυτά τα ερωτήματα ήταν:

- Εύρεση παίχτη με νίκες μετά από ήττα σε πρώτο (ή επόμενο) σετ
- Ανάλυση στο Εύρος ηλικιών των νικητών των Grand Slam τουρνουά

- Ανάλυση στο Εύρος ηλικιών του top 10 ανά τα χρόνια
- Ανάλυση στο Εύρος ηλικιών του top 10 ανά τα χρόνια
- Ανάλυση στις εθνικότητες του top 10 ανά τα χρόνια
- Ποσοστό σε νίκες μετά σε Grand Slam αγώνων μετά από ήττα σε πρώτο (ή επόμενο) σετ
- Ποσοστό νικών μετά από ήττα σε δεύτερο σετ
- Εύρεση μέγιστων διαφορών ύψους και ηλικίας μεταξύ των αγώνων
- Εύρεση των μεγαλύτερων εκπλήξεων βάσει των διαφορών στο ranking
- Εύρεση μεγαλύτερης καριέρας σε διάρκεια
- Εύρεση παικτών που δεν έχουν αποσυρθεί από κανένα αγώνα, καθώς και των νικών τους.

Οι πληροφορίες αυτές μπορούν να είναι χρήσιμες για μία πιθανή επέκταση της εργασίας.

4.2 Συμπεράσματα και τρόποι χρήσης της EDA

Ένα πιθανό ερώτημα είναι το πως μπορεί να γίνει χρήση των πληροφοριών αυτών, αλλά και γενικότερα πως κάποιος θα μπορούσε να κάνει χρήση του exploratory data analysis στο τένις. Ορισμένες περιπτώσεις είναι οι κάτωθι:

1. **Προπονητικές υπηρεσίες:** Με τη χρήση της ανάλυσης δεδομένων, μπορούν να εντοπιστούν διάφοροι τομείς παιχνιδιού οι οποίοι μπορύν να βελτιωθούν από τους προπονητές. Έτσι, ως μέρος της προπόνησης, θα μπορούσε να γίνει χρήση του EDA ώστε να υπάρχει μία επιπλέον υπηρεσία για τους αθλούμενους, η οποία θα προσφέρεται σε παίκτες, ή και προπονητές, με σκοπό τη βελτίωση της απόδοσης. Η υπηρεσία αυτή, θα μπορούσε να προσφέρει επιπλέον έσοδα φυσικά, καθώς απαιτεί ειδικές γνώσεις και μπορούν έτσι να αναπτυχθούν και ατομικευμένα προγράμματα ασκήσεων, με σκοπό τη βελτίωση τομέων του παιχνιδιού.

2. **Δημιουργία προϊόντων/Υπηρεσιών:** Με τη χρήση ανάλυσης δεδομένων, υπάρχει η δυνατότητα δημιουργίας νέων προϊόντων ή και της βελτίωσης των υπάρχοντων προϊόντων(για παράδειγμα, οι μπάλες του γκόλφ, έχουν εξελιχθεί ιδιαιτέρως με τα χρόνια). Τα νέα προϊόντα αυτά θα δημιουργηθούν έχοντας ως γνώμονα τον χρήστη, και έτσι θα βοηθούν στη βελτιστοποίηση της απόδοσης. Με τη χρήση του EDA, μπορούν συνεπώς να αναπτυχθούν προϊόντα που έχουν μεγαλύτερη πιθανότητα να επιτύχουν. Αυτό θα συμβεί διότι θα γίνει προτίστως ανάλυση των αναγκών και των προτιμήσεων των χρηστών, και έτσι το προϊόν θα προσφέρει λύση σε μία η περισσότερες ανάγκες.

3. **Παροχή συμβουλευτικών υπηρεσιών:** Επιπλέον των προϊόντων, μπορούν να δοθούν και υπηρεσίες για προτάσεις και εξειδικευμένες πληροφορίες. Οι υπηρεσίες αυτές θα δίνονται σε κατασκευαστές, για παράδειγμα, εξοπλισμού ή και διοργανωτές αγώνων. Οι εξειδικευμένες αυτές υπηρεσίες θα αμοίβονται και μπορούν να δημιουργήσουν, ενδεχομένως, επιπλέον έσοδα.

4. **Έρευνα και ανάπτυξη προϊόντων:** Με τη χρήση της EDA, μπορεί επίσης να γίνει και διαδικασία έρευνας για το τένις. Ως ερευνητής, κάποιος θα μπορούσε να ακολουθήσει μία καριέρα σε πανεπιστημιακό ή και επιχειρηματικό επίπεδο, πραγματοποιώντας έρευνα ως ακαδημαϊκός ή ως σύμβουλος. Έτσι, ανακαλύψεις όπως νέες μετρικές, ιδιαίτεροι μέθοδοι ή και τεχνικές ανάλυσης, μπορούν να δημοσιευθούν σε επιστημονικά περιοδικά και συνέδρια, ως μέρος μίας καριέρας σε αυτό τον τομέα.

5. **Χρήση για υπηρεσίες στοιχηματισμού:** Ένας ακόμη τρόπος για τη χρήση του EDA είναι επίσης και η χρήση του για υπηρεσίες στοιχηματισμού, ώστε να δημιουργηθεί μία στρατηγική ποντάρισματος. Μία τέτοια στρατηγική θα μπορούσε να βασιστεί σε δεδομένα που εξάγονται με τη χρήση του EDA και να βοηθήσει στην επιλογή του ποντάρισματος. Οι επιλογές αυτές θα προκύπτουν με βάση την ανάλυση και έτσι θα εμπνέουν περισσότερη εμπιστοσύνη αφού μπορεί να προκύπτουν από ανάλυση δεδομένων παικτών, αγώνων, ή και το συνδευασμό αυτών. Τα μοτίβα που θα εξαχθούν θα δίνουν μεγαλύτερη σιγουριά και έχουν υψηλότερες πιθανότητες για επιτυχία, σε σύγκριση με το κλασσικό ποντάρισμα (το οποίο, σε κάθε περίπτωση, πρέπει να γίνεται υπεύθυνα).

Συνεπώς, μπορεί να γίνει κατανοητό, ότι η χρήση των δεδομένων και η εφαρμογή του exploratory data analysis, μπορεί να δώσει επιπλέον αξία και να βοηθήσει ιδιαίτερα σε διάφορους τομείς. Ο χρήστης ωστόσο είναι πολύ σημαντικός, ώστε να επιλέξει τις πληροφορίες

που θα κρατήσει, καθώς και να τις χρησιμοποιήσει με τρόπο υπεύθυνο, αλλά και όπου αυτό χρειάζεται, ηθικό.

Bibliography

[1] *18 Data Science Tools to Consider Using in 2023*. https://www.techtarget.com/searchbusinessanalytics/feature/15-data-science-tools-to-consider-using. (Accessed on 05/30/2023).

[2] Robert St. Amant and Paul R Cohen. "Intelligent Support for Exploratory Data Analysis". In: *Journal of Computational and Graphical Statistics* 7.4 (1998), pp. 545–558. DOI: 10.1080/10618600.1998.10474794. URL: https://doi.org/10.1080/10618600.1998.10474794.

[3] Jonathan D Becher, Pavel Berkhin, and Edmund Freeman. "Automating Exploratory Data Analysis for Efficient Data Mining". In: (2000).

[4] Jerome H. Friedman and John W. Tukey. "A Projection Pursuit Algorithm for Exploratory Data Analysis". In: *IEEE Transactions on Computers* C-23.9 (1974), pp. 881–890. ISSN: 00189340. DOI: 10.1109/T-C.1974.224051.

[5] Mark Jamison. *tennisim · PyPI*. https://pypi.org/project/tennisim/. (Accessed on 05/29/2023).

[6] Marjan Kuchaki Rafsanjani, Zahra Asghari Varzaneh, and Nasibeh Emami Chukanlo. "A Survey Of Hierarchical Clustering Algorithms". In: *Journal of Mathematics and Computer Science* 05.03 (Oct. 2012), pp. 229–240. DOI: 10.22436/JMCS.05.03.11. URL: https://www.isr-publications.com/jmcs/articles-417-a-survey-of-hierarchical-clustering-algorithms.

[7] Xutao Li et al. "On cluster tree for nested and multi-density data clustering". In: *Pattern Recognition* 43.9 (Sept. 2010), pp. 3130–3143. ISSN: 0031-3203. DOI: 10.1016/J.PATCOG.2010.03.020.

[8] J B MacQueen. "Some Methods for Classification and Analysis of MultiVariate Observations". In: *Proc. of the fifth Berkeley Symposium on Mathematical Statistics and Probability*. Ed. by L M Le Cam and J Neyman. Vol. 1. University of California Press, 1967, pp. 281–297.

[9] S K Mukhiya and U Ahmed. *Hands-On Exploratory Data Analysis with Python: Perform EDA techniques to understand, summarize, and investigate your data*. Packt Publishing, 2020. ISBN: 9781789535624. URL: https://books.google.be/books?id=QcHZDwAAQBAJ.

[10] Glenn J. Myatt and Wayne P. Johnson. "Making sense of data I : a practical guide to exploratory data analysis and data mining". In: (). URL: https://www.wiley.com/en-be/Making+Sense+of+Data+I%3A+A+Practical+Guide+to+Exploratory+Data+Analysis+and+Data+Mining%2C+2nd+Edition-p-9781118407417.

[11] A Pajankar. *Practical Python Data Visualization: A Fast Track Approach To Learning Data Visualization With Python*. Apress, 2021. ISBN: 9781484267516. URL: https://books.google.be/books?id=XatizgEACAAJ.

[12] Aggarwal • Reddy. "DATA CLUSTERING DATA CLUSTERING Algorithms and Applications Chapman & Hall/CRC Data Mining and Knowledge Discovery Series Chapman & Hall/CRC Data Mining and Knowledge Discovery Series". In: (2014).

[13] Mel Restori. *What is Exploratory Data Analysis | Tutorial by Chartio*. 2019. URL: https://chartio.com/learn/data-analytics/what-is-exploratory-data-analysis/ (visited on 01/17/2023).

[14] Pablo Roca. *ATP - Tennis Matches - 2000-2019 | Kaggle*. https://www.kaggle.com/datasets/pablodroca/atp-tennis-matches-20002019. (Accessed on 05/31/2023).

[15] David W. Scott. "Multivariate Density Estimation : Theory, Practice, and Visualization." In: (2015), p. 381.

[16] Alexander De Seranno. *Predicting Tennis Matches Using Machine Learning*. https://libstore.ugent.be/fulltxt/RUG01/002/945/727/RUG01-002945727_2021_0001_AC.pdf. (Accessed on 05/29/2023).

[17] Michal Sipko. *Machine Learning for the Prediction of Professional Tennis Matches*. https://www.doc.ic.ac.uk/teaching/distinguished-projects/2015/m.sipko.pdf. (Accessed on 05/29/2023).

[18] John W Tukey. *Exploratory Data Analysis*. Addison-Wesley, 1977.

[19] Karan Vombatkere. *Predicting Professional Tennis Player Success using Binary Classification Methods*. https://kvombatkere.github.io/assets/Vombatkere_TennisPlayerPrediction_WriteUp.pdf. (Accessed on 05/29/2023).

[20] Jack C. Yue et al. “A study of forecasting tennis matches via the Glicko model”. In: *PLOS ONE* 17.4 (Apr. 2022), pp. 1–12. DOI: 10.1371/journal.pone.0266838. URL: https://doi.org/10.1371/journal.pone.0266838.

[21] *Η ιστορία των Ολυμπιακών Αγώνων - Αφιέρωμα - Σαν Σήμερα .gr*. https://www.sansimera.gr/articles/545. (Accessed on 04/25/2023).

[22] *Το θρησκευτικό στοιχείο στους Ολυμπιακούς αγώνες κατά την αρχαιότητα*. http://ikee.lib.auth.gr/record/67621/files/Plakoti.pdf. (Accessed on 04/25/2023).