Πανεπιστήμιο Θεσσαλίας
Πολυτεχνική Σχολή
Τμήμα Ηλεκτρολόγων Μηχανικών & Μηχανικών Υπολογιστών

Αναλυτική Επεξεργασία Δεδομένων Αθλητικών Αγωνισμάτων

Analytical Processing of Sports Data

Διπλωματική Εργασία

ΠΕΤΡΟΣ Δ. ΞΕΣΦΙΓΓΗΣ

Επιβλέπων
Μιχαήλ Βασιλακόπουλος
Αναπληρωτής Καθηγητής

Βόλος, Δεκέμβριος 2018

Πανεπιστήμιο Θεσσαλίας
Πολυτεχνική Σχολή
Τμήμα Ηλεκτρολόγων Μηχανικών & Μηχανικών Υπολογιστών

Αναλυτική Επεξεργασία Δεδομένων Αθλητικών Αγωνισμάτων

Analytical Processing of Sports Data

Διπλωματική Εργασία

ΠΕΤΡΟΣ Δ. ΞΕΣΦΙΓΓΗΣ

Επιτροπή επίβλεψης

Επιβλέπων
Μιχαήλ Βασιλακόπουλος
Αναπληρωτής Καθηγητής

Συνεπιβλέπουσα
Ελένη Τουσίδου
Ε.ΔΙ.Π.

Βόλος, Δεκέμβριος 2018

Πανεπιστήμιο Θεσσαλίας
Πολυτεχνική Σχολή
Τμήμα Ηλεκτρολόγων Μηχανικών & Μηχανικών Υπολογιστών

Ο συγγραφέας αυτής της εργασίας βεβαιώνει ότι κάθε βοήθεια την οποία είχε για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης βεβαιώνει ότι έχει αναφέρει τις όποιες πηγές από τις οποίες έκανε χρήση δεδομένων, ιδεών ή λέξεων, είτε αυτές αναφέρονται επακριβώς, είτε παραφρασμένες.

Περίληψη

Τα τελευταία χρόνια έχει αναπτυχθεί ιδιαίτερα ο τομέας της αναλυτικής αθλητικών αγωνισμάτων, δηλαδή της χρήσης τεχνικών της επιστήμης δεδομένων για την εξαγωγή συμπερασμάτων που μπορούν να συνεισφέρουν στην στήριξη των αποφάσεων αθλητικών συλλόγων ή οργανισμών. Η παρούσα διπλωματική εργασία εξετάζει δεδομένα από το Παγκόσμιο Κύπελλο Ποδοσφαίρου του 2018 με την προσέγγιση της άμεσης αναλυτικής επεξεργασίας (OLAP), δηλαδή εκείνη της πολυδιάστατης εξέτασης μιας αποθήκης δεδομένων. Αποθήκη δεδομένων ονομάζεται μια βάση δεδομένων, συνήθως μεγάλου μεγέθους, που περιέχει ιστορικά δεδομένα και χρησιμοποιείται για την αναφορά και την ανάλυση τους. Για την πραγματοποίηση της παραπάνω ανάλυσης δεδομένων έχει δημιουργηθεί μια εφαρμογή σε προγραμματιστικό περιβάλλον Python, η οποία στηρίζεται στο εργαλείο Cubes. Στην εργασία παρουσιάζεται εκτενώς η αρχιτεκτονική και ο τρόπος χρήσης του εν λόγω εργαλείου και περιγράφονται αναλυτικά όλα τα απαραίτητα βήματα για την δημιουργία της εφαρμογής. Στην συνέχεια, παρατίθενται κάποια από τα πιο ενδεικτικά ερωτήματα που μπορεί να απαντήσει το σύστημα για το Παγκόσμιο Κύπελλο του 2018 και τα βασικά συμπεράσματα που προκύπτουν από την ανάλυση. Ο χρήστης μπορεί να χρησιμοποιήσει την εφαρμογή, μετά από την λήψη της, για να υποβάλλει τα δικά του/της ερωτήματα και να αντλήσει τις πληροφορίες για τις οποίες ενδιαφέρεται.

Λέξεις Κλειδιά

Αποθήκες Δεδομένων, Άμεση Αναλυτική Επεξεργασία, Αθλητικά Δεδομένα, Python, Cubes Framework

Abstract

Sports Analytics, namely the use of data science techniques to drawn information that can be useful to support decision making from sports clubs or organizations, have been developed drastically in recent years to the point that they have played a significant role in major changes. The current thesis analyzes data from FIFA World Cup of 2018 with the approach of Online Analytical Processing (OLAP) that is multi-dimensional analysis of data warehouses. A data warehouse is a, usually large, database that holds historical data for reporting and analysis. To achieve the Online Analytical Processing for the World Cup, an application has been created in Python environment using the Cubes framework. The thesis presents the architecture of the abovementioned framework and serves as a user's guide for Cubes. Additionally, it describes all the necessary steps for the creation not only of the application included but also for a Python OLAP application in general. Finally, some of the most indicative queries that the system can respond to are presented and the main conclusions drawn from the data analysis are discussed. The reader can download the application to submit his/her own queries and gain the information he/she is interested in.

Keywords

Data Warehouses, OLAP, Sports Data, Python, Cubes Framework

Ευχαριστίες

Στο σημείο αυτό θα ήθελα να ευχαριστήσω θερμά τον καθηγητή κ. Μιχαήλ Βασιλακόπουλο για τις καίριες συμβουλές και την καθοδήγηση που μου παρείχε σε όλα τα στάδια της εκπόνησης της παρούσας διπλωματικής εργασίας. Επίσης θα ήθελα να ευχαριστήσω τους γονείς μου για την υποστήριξη που μου προσέφεραν κατά την διάρκεια των σπουδών μου.

Πρόλογος

Η παρούσα εργασία εκπονήθηκε ως το τελευταίο βήμα για την απόκτηση διπλώματος και την ολοκλήρωση των σπουδών μου στο τμήμα Ηλεκτρολόγων Μηχανικών & Μηχανικών Υπολογιστών του Πανεπιστημίου Θεσσαλίας στην πόλη του Βόλου υπό την επίβλεψη του καθηγητή κ. Μιχαήλ Βασιλακόπουλου.

Περιεχόμενα

Κατάλογος σχημάτων

Κατάλογος πινάκων

Κεφάλαιο 1

Εισαγωγή

Οι τεχνολογικές εξελίξεις των τελευταίων δεκαετιών έχουν επιτρέψει την συλλογή και την αποθήκευση τεράστιου όγκου δεδομένων που μάλιστα αναμένεται να αυξηθεί εκθετικά μέσα στα επόμενα χρόνια με την ανάπτυξη σε τομείς όπως των πολυμέσων, των κοινωνικών δικτύων και του διαδικτύου των πραγμάτων. Σύντομα το πρόβλημα έπαψε να είναι η εύρεση των δεδομένων αλλά η εύρεση τρόπων για την αποδοτική αξιοποίηση τους, για τον λόγο αυτό άρχισε να εμφανίζεται στο προσκήνιο η επιστήμη των δεδομένων η οποία συνδυάζει την πληροφορική, την στατιστική, την μηχανική μάθηση, την οπτικοποίηση και την αλληλεπίδραση ανθρώπου – υπολογιστή με σκοπό την συλλογή, την επεξεργασία και την ανάλυση των δεδομένων. Τα τελευταία χρόνια η επιστήμη των δεδομένων έχει εισχωρήσει δυναμικά σε όλους τους τομείς της επιστήμης και της οικονομίας καθώς μπορεί να οδηγήσει στην άντληση χρήσιμων πληροφοριών και μοτίβων που βρίσκονται «κρυμμένες» μέσα σε βάσεις δεδομένων, πληροφορίες που μπορούν να συνεισφέρουν καθοριστικά κατά την διαδικασία της λήψης των αποφάσεων. [3, 5]

1.1 Αντικείμενο της διπλωματικής

Αντικείμενο της παρούσας διπλωματικής εργασίας αποτελεί η άμεση αναλυτική επεξεργασία (OLAP) δεδομένων ποδοσφαιρικών αγώνων για την εξαγωγή γνώσης και χρήσιμων συμπερασμάτων που μπορούν να αξιοποιηθούν από παίκτες του στοιχήματος, από ποδοσφαιρικούς αναλυτές και κυνηγούς ταλέντων αλλά και από απλούς λάτρεις του αγωνίσματος. Για τον λόγο αυτό θα χρησιμοποιηθεί το εργαλείο Cubes, το οποίο πρόκειται για ένα framework υλοποιημένο στην προγραμματιστική γλώσσα Python σε συνδυασμό με έναν ενσωματωμένο HTTP Server, που επιτρέπει σε εφαρμογές Python να πραγματοποιούν τις κυριότερες εργασίες που συνθέτουν την άμεση αναλυτική επεξεργασία. Επίσης, το Cubes μπορεί να υποστηρίξει πληθώρα διαφορετικών βάσεων δεδομένων τόσο σχεσιακές όσο και μη-σχεσιακές και καθίσταται κατά αυτόν τον τρόπο ιδιαίτερα προσαρμόσιμο και λειτουργικό ανεξάρτητα με την μορφή που έχουν τα δεδομένα. Η διπλωματική εργασία θα έχει και χαρακτήρα οδηγού για το Cubes παρουσιάζοντας την δομή και την λειτουργία του καθώς επίσης και οδηγίες για την εγκατάσταση και την χρήση του από κάποιον αρχάριο.

Έχει δημιουργηθεί σετ δεδομένων (dataset) το οποίο περιλαμβάνει 736 στοιχεία και 14 γνωρίσματα και αφορά τους ποδοσφαιριστές που συμμετείχαν στο Παγκόσμιο Κύπελλο του 2018, περι-

λαμβάνει γενικές πληροφορίες καθώς και στατιστικά στοιχεία από την παρουσία του καθενός στην διοργάνωση. Το σετ δεδομένων αυτό μοντελοποιείται ως υπερκύβος ο οποίος μπορεί να «προσπελαστεί» με τη χρήση του Cubes από την εφαρμογή που θα αναπτυχθεί η οποία μέσω γραφικού περιβάλλοντος θα δέχεται από τον χρήστη τα ερωτήματα και θα οπτικοποιεί τα αποτελέσματα με κατάλληλα γραφήματα. Στην συνέχεια θα παρουσιαστούν αναλυτικά τα συμπεράσματα και η κρυμμένη πληροφορία που θα εξαχθεί από τα δεδομένα με την χρήση της εφαρμογής.

1.1.1 Συνεισφορά

Η συνεισφορά της διπλωματικής συνοψίζεται ως εξής:

1. Παρουσιάζεται η ιστορική εξέλιξη του τομέα των Sports Analytics
2. Επεξηγούνται οι βασικές έννοιες της άμεσης αναλυτικής επεξεργασίας
3. Παρουσιάζεται αναλυτικά η αρχιτεκτονική και ο τρόπος χρήσης του εργαλείου Cubes
4. Δημιουργήθηκε βάση δεδομένων που αφορά τους ποδοσφαιριστές που συμμετείχαν στο Παγκόσμιο Κύπελλο του 2018
5. Παρουσιάζονται όλα τα βήματα για την προετοιμασία της συγκεκριμένης βάσης δεδομένων έτσι ώστε να μπορεί να αναλυθεί με το εργαλείο Cubes
6. Υλοποιείται εφαρμογή σε γλώσσα Python για την άμεση αναλυτική επεξεργασία της βάσης δεδομένων αυτής και της οπτικοποίησης των αποτελεσμάτων
7. Παρουσιάζονται τα κύρια συμπεράσματα που προέκυψαν από την ανάλυση

1.2 Οργάνωση του τόμου

Στο Κεφάλαιο 2 παρουσιάζεται το θεωρητικό υπόβαθρο της διπλωματικής εργασίας με την περιγραφή και ιστορική εξέλιξη του τομέα της αναλυτικής αθλητικών αγωνισμάτων (Sports Analytics) καθώς και την λεπτομερή περιγραφή των βασικών εννοιών της άμεσης αναλυτικής επεξεργασίας (OLAP). Η λεπτομερής περιγραφή του εργαλείου Cubes καθώς και οδηγίες για την εγκατάσταση και την χρήση του από κάποιον αρχάριο αναπτύσσονται στο Κεφάλαιο 3. Το Κεφάλαιο 4 ασχολείται με την δημιουργία της βάσης δεδομένων και την υλοποίηση της εφαρμογής σε προγραμματιστικό περιβάλλον Python ή οποία με την χρήση του Cubes θα επιτυγχάνει την άμεση αναλυτική επεξεργασία των δεδομένων. Τέλος το Κεφάλαιο 5 συζητά τα κύρια συμπεράσματα που εξάγονται σε αυτή την διπλωματική με την χρήση της εφαρμογής που υλοποιήθηκε.

Κεφάλαιο 2

Θεωρητικό υπόβαθρο

2.1 Εισαγωγή

Πριν από την παρουσίαση της ανάλυσης και της σχεδίασης του συστήματος, είναι σημαντικό να γίνει κατανοητό το πως σχετίζεται η επιστήμη των δεδομένων με τον αθλητισμό και ποια είναι τα οφέλη που μπορούν να προκύψουν από την χρήση αυτής για όλους τους άμεσα ή έμμεσα εμπλεκόμενους με την αθλητική βιομηχανία. Επίσης καθώς η εργασία θα ασχοληθεί με την άμεση αναλυτική επεξεργασία, δηλαδή το σύνολο των εργασιών που σχετίζονται με τις αποθήκες δεδομένων, είναι σημαντικό για τον αναγνώστη να κατανοήσει τις βασικές έννοιες και την ορολογία που χρησιμοποιείται προτού προχωρήσει στα επόμενα κεφάλαια.

2.2 Αναλυτική Αθλητικών Αγωνισμάτων (Sports Analytics)

Τα τελευταία χρόνια υπάρχει εκτενής χρήση της στατιστικής και της επιστήμης των δεδομένων στον αθλητισμό. Ο συγκεκριμένος τομέας άρχισε να αναπτύσσεται στις αρχές της προηγούμενης δεκαετίας όταν ο αθλητικός διευθυντής, της αμερικάνικης ομάδας μπέιζμπολ Oakland Athletics, Billy Beane επιδίωξε να δημιουργήσει μια ανταγωνιστική ομάδα έχοντας σημαντικά χαμηλότερο προϋπολογισμό από τους αντιπάλους του χρησιμοποιώντας σε μεγάλο βαθμό στατιστικές και μεθόδους εξόρυξης δεδομένων και μηχανικής μάθησης για την επιλογή του συστήματος και την στελέχωση της ομάδας. Τα εντυπωσιακά αποτελέσματα της πρακτικής αυτής φέρανε την επιστήμη των δεδομένων στο προσκήνιο του παγκόσμιου αθλητισμού με όλο και περισσότερους αθλητικούς συλλόγους να χρησιμοποιούν τέτοιες μεθόδους για την στήριξη των αποφάσεων τους σχετικά με την επιλογή των αθλητών, των προπονητών και συνολικότερα τον τρόπο με τον οποίο αγωνίζεται η ομάδα, αποφάσεις που μέχρι τότε βασίζονταν καθαρά στο ένστικτο και την εμπειρία του αρμόδιου επιτελείου. Σήμερα όλοι οι μεγάλοι αθλητικοί όμιλοι διαθέτουν αρμόδιο προσωπικό για την συλλογή και αξιοποίηση δεδομένων με στόχο την στήριξη των αποφάσεων έτσι ώστε να αποκτήσουν συγκριτικό πλεονέκτημα έναντι στους ανταγωνιστές τους. [8, 7, 4]

Ο τομέας της αναλυτικής αθλητικών αγωνισμάτων διακρίνεται σε δύο μεγάλες κατηγορίες, εντός γηπέδου (on-field) και εκτός γηπέδου (off-field). Στην πρώτη κατηγορία εντάσσονται όλες οι τεχνικές που έχουν ως στόχο την βελτίωση των επιδόσεων της ομάδας ή συγκεκριμένων αθλητών

αυτής. Τέτοιες τεχνικές επιδιώκουν να κατανοήσουν ποιοι αθλητές αποδίδουν καλά όταν αγωνίζονται μαζί, πώς ανταποκρίνονται απέναντι σε αντιπάλους με συγκεκριμένα χαρακτηριστικά, ποιο είναι το κατάλληλο σύστημα παιχνιδιού για έναν συγκεκριμένο συνδυασμό παικτών ενώ ταυτόχρονα προσπαθούν να προβλέψουν και τον τρόπο με τον οποίο αγωνίζονται οι αντίπαλες ομάδες έτσι ώστε να υπάρχει η κατάλληλη προετοιμασία. Περαιτέρω ένας ακόμα τομέας στον οποίο προσπαθεί να προσφέρει η επιστήμη των δεδομένων στον αθλητισμό είναι αυτός της πρόληψης και αποφυγής τραυματισμών. [7, 4]

Η τεχνολογία παίζει μεγάλο ρόλο στην συλλογή των δεδομένων έτσι ώστε να μπορούν να πραγματοποιηθούν τέτοιες αναλύσεις. Στο Παγκόσμιο Κύπελλο που διεξήχθη στην Βραζιλία το 2014, οι ποδοσφαιριστές της εθνικής ομάδας της Γερμανίας κατά την διάρκεια των προπονήσεων φορούσαν ένα ειδικό σύστημα αισθητήρων που κατέγραφε τον καρδιακό τους ρυθμό, την επιτάχυνση, την ταχύτητα και την απόσταση που διένυαν έτσι ώστε να μπορεί να καθοριστεί από τα δεδομένα αυτά η φόρμα στην οποία βρισκόταν ο κάθε ποδοσφαιριστής και να γίνει καλύτερη διαχείριση του αγωνιστικού τους χρόνου. Παρόμοια συστήματα χρησιμοποιούνται πλέον από όλους τους μεγάλους συλλόγους ενώ υπάρχει και μεγάλο πλήθος εταιριών που σχεδιάζουν τέτοια συστήματα καταγραφής. Οι ομάδες του αμερικάνικου πρωταθλήματος καλαθοσφαίρισης (NBA) χρησιμοποιούν ένα σύστημα που ονομάζεται Player Tracking το οποίο χρησιμοποιεί μια σειρά από κάμερες οι οποίες παρακολουθούν τις κινήσεις όλων των αθλητών που βρίσκονται στον αγωνιστικό χώρο 25 φορές το δευτερόλεπτο και συλλέγουν έτσι δεδομένα για την κίνηση και τις ενέργειες τους που οδηγούν σε πληθώρα στατιστικών δεικτών. [8, 7]

Σχήμα 2.1: Sports Analytics [4]

Η δεύτερη μεγάλη κατηγορία είναι αυτή που εστιάζει στην εκτός γηπέδου δυναμική των ομάδων. Όπως είναι γνωστό ο επαγγελματικός αθλητισμός την σύγχρονη εποχή είναι βιομηχανία δισεκατομμυρίων και οι ομάδες είναι ταυτόχρονα και μεγάλες εταιρίες. Επομένως η εκτός γηπέδου αναλυτική αθλητικών αγωνισμάτων ασχολείται με αυτήν την αναπόσπαστη διάσταση των ομάδων και συνεισφέρει στον καθορισμό των τιμών των εισιτηρίων, την προώθηση του συλλόγου έτσι ώστε να αποκτήσει περισσότερους φιλάθλους, τις πωλήσεις των προϊόντων που φέρουν το έμβλημα του συλλόγου και γενικότερα όλα όσα αφορούν την μεγιστοποίηση του κέρδους της εταιρίας. Ακόμα και η επιλογή των αθλητών πολλές φορές δεν γίνεται καθαρά με αγωνιστικά κριτήρια αλλά και με βάση τα προσδοκώμενα έσοδα που μπορεί να φέρει η δημοφιλία συγκεκριμένων αθλητών. [2, 4]

Για τους παραπάνω λόγους οι ομάδες συλλέγουν δεδομένα για τις προτιμήσεις των φιλάθλων μέσω του διαδικτύου και των μέσων κοινωνικής δικτύωσης έτσι ώστε να κατανοήσουν καλύτερα τις επιθυμίες τους και να επιτύχουν βαθύτερη σύνδεση μαζί τους. Έχουν παρατηρηθεί περιπτώσεις ομάδων που αν και δεν σημειώνουν σημαντικές επιτυχίες εντός των γηπέδων, βρίσκονται πολύ ψηλά στις κατατάξεις των πωλήσεων εισιτηρίων και προϊόντων και ταυτόχρονα έχουν μεγάλο και αυξανόμενο αριθμό φιλάθλων. Αντίθετα ομάδες που σημειώνουν αγωνιστικές επιτυχίες δύναται να μην τα καταφέρνουν εξίσου καλά στο οικονομικό κομμάτι. Συνεπώς η χρήση της επιστήμης των δεδομένων αποσκοπεί στην εύρεση των παραγόντων εκείνων που κάνουν τις ομάδες να κερδοφορούν και να αυξάνουν την δημοφιλία τους πέρα από τις αγωνιστικές τους επιδόσεις. [2]

Η έξαρση στην δημοτικότητα της χρήσης της επιστήμης δεδομένων στον επαγγελματικό αθλητισμό έχει μεγάλη απήχηση και στο φίλαθλο κοινό, συμπέρασμα που προκύπτει από την πληθώρα ιστοσελίδων με σημαντική απήχηση που ασχολούνται με την καταγραφή και την παρουσίαση αναλυτικών στατιστικών αθλητικών αγωνισμάτων. Τέλος ένας ακόμα τομέας που σχετίζεται άμεσα είναι αυτός του στοιχήματος σε αθλητικά γεγονότα. Οι παίκτες του στοιχήματος χρησιμοποιούν αναλυτικά στατιστικά, από πολύ απλά έως πολύ σύνθετα, έτσι ώστε να μπορούν να προβλέψουν καλύτερα τα αποτελέσματα των αγώνων στους οποίους επιθυμούν να στοιχηματίσουν και να βελτιώσουν τις πιθανότητες τους για κέρδος.

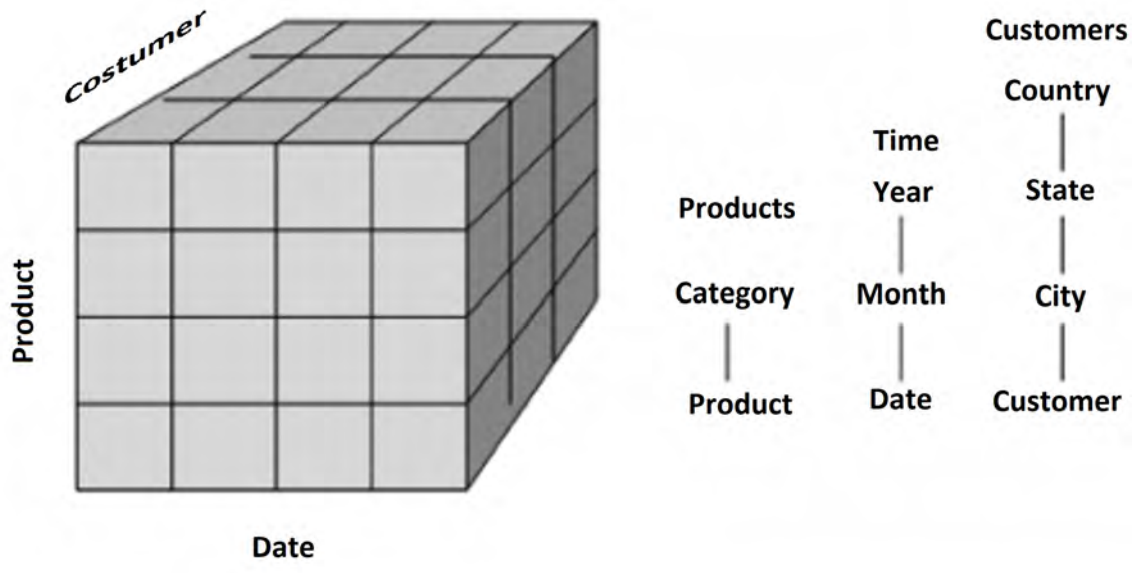
Για την επίτευξη όσων αναφέρθηκαν προηγουμένως και την εξαγωγή καίριων συμπερασμάτων με σκοπό την αποτελεσματική στήριξη των αποφάσεων απαιτείται η δημιουργία μεγάλων και κατάλληλα οργανωμένων αποθηκών δεδομένων. Μια αποθήκη δεδομένων είναι μια βάση δεδομένων, συνήθως μεγάλου μεγέθους, που περιλαμβάνει αμετάβλητα ιστορικά δεδομένα για αναφορά και ανάλυση με σκοπό την στήριξη αποφάσεων. Οι κύριες εργασίες ενός συστήματος αποθήκης δεδομένων με τις οποίες επιτυγχάνεται η ανάλυση των δεδομένων που οδηγεί στην εξαγωγή των συμπερασμάτων είναι γνωστές με τον όρο Άμεση Αναλυτική Επεξεργασία (OLAP).

2.3 Άμεση Αναλυτική Επεξεργασία (OLAP)

Η άμεση αναλυτική επεξεργασία (OLAP) αποτελεί το σύνολο των εργασιών που συνήθως σχετίζονται με μια αποθήκη δεδομένων, συνεπώς δεν αφορούν την τροποποίηση των δεδομένων αλλά

την ανάγνωσή τους με έναν πιο ολοκληρωμένο και πολύπλοκο τρόπο έτσι ώστε να μπορούν να αντληθούν χρήσιμες πληροφορίες που μπορούν να στηρίξουν την λήψη αποφάσεων. Τα συστήματα άμεσης αναλυτικής επεξεργασίας εστιάζουν στην εξαγωγή στατιστικών στοιχείων από τα δεδομένα και την οπτικοποίηση τους ενώ διέπονται από έναν αλληλεπιδραστικό χαρακτήρα προβάλλοντας στον χρήστη το υποσύνολο της πληροφορίας για το οποίο ενδιαφέρεται. [9, 14]

Τα συστήματα OLAP βασίζονται στην πολυδιάστατη αναπαράσταση των δεδομένων, πυρήνας της οποίας αποτελεί ένας πίνακας που περιλαμβάνει το σύνολο των Facts δηλαδή των εγγραφών του χαρακτηριστικού που θα τεθεί ως το επίκεντρο της ανάλυσης, ο πίνακας αυτός ονομάζεται πίνακας γεγονότων (Fact Table). Παραδείγματος χάριν, Fact μπορεί να είναι μια αγοραπωλησία, ένα συμβόλαιο ή ένας υπάλληλος. Επίσης, απαιτείται ο προσδιορισμός των διαστάσεων (π.χ. εάν στο επίκεντρο της ανάλυσης βρίσκεται μια αγοραπωλησία, ως διαστάσεις μπορούν να τεθούν η ημερομηνία και ο χώρος στον οποίο διεξήχθη). Κατά αυτόν τον τρόπο, κάθε συνδυασμός των τιμών των χαρακτηριστικών που έχουν τεθεί ως διαστάσεις αντιστοιχεί σε ένα κελί της πολυδιάστατης αναπαράστασης. [9, 14]

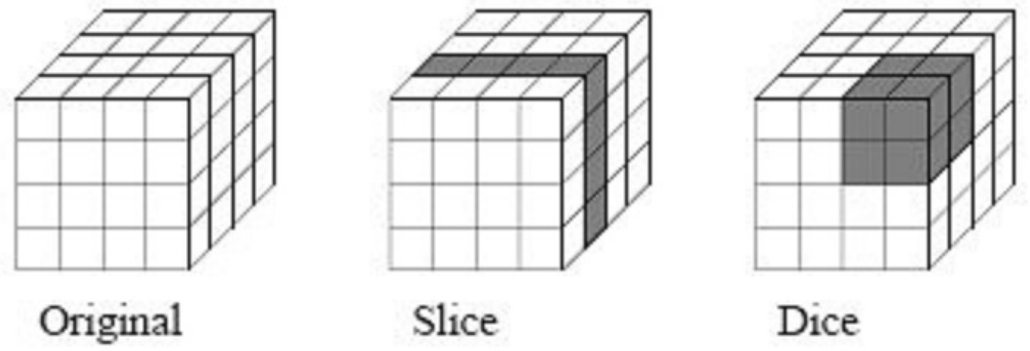


Σχήμα 2.2: Παράδειγμα κύβου δεδομένων τριών διαστάσεων [6]

Για ένα fact συνήθως προσδιορίζονται συγκεκριμένα μέτρα (measures), δηλαδή χαρακτηριστικά που περιγράφουν ποσοτικά το γεγονός. Μια αγοραπωλησία για παράδειγμα μπορεί να έχει ως μέτρα την τιμή στην οποία πραγματοποιήθηκε και τον φόρο που της αντιστοιχεί. Ένα κελί θα περιέχει ορισμένες συνολικές τιμές ή συναθροίσεις (aggregates), όπου τέτοιες μπορεί να είναι το πλήθος των γεγονότων που εμπίπτουν στις τιμές των διαστάσεων που έχουν ορίσει το συγκεκριμένο κελί (π.χ. το πλήθος των αγοραπωλησιών) ή συναθροίσεις που αφορούν συγκεκριμένα μέτρα (measures) του γεγονότος. [9, 14]

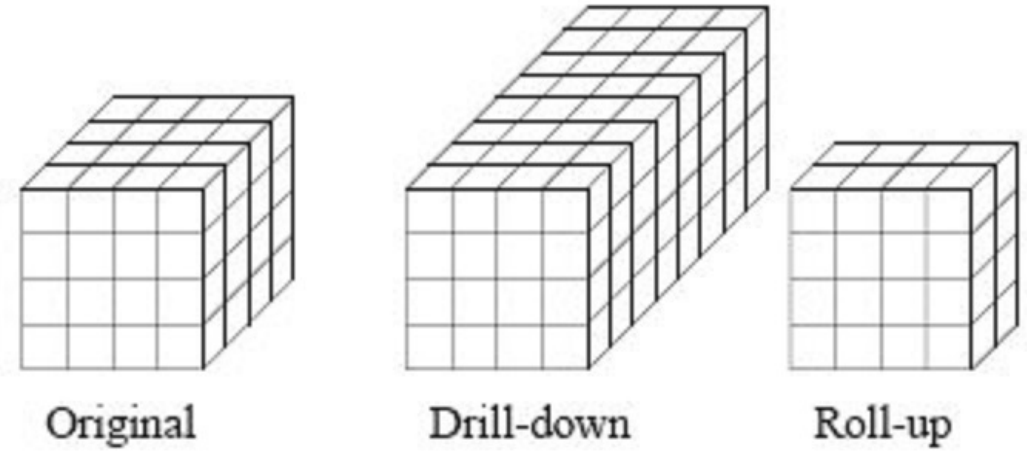
Μια πολυδιάστατη αναπαράσταση των δεδομένων με τα χαρακτηριστικά που περιγράφηκαν παραπάνω μαζί με όλες τις συναθροίσεις που έχουν οριστεί για αυτή ονομάζεται κύβος δεδομένων (data cube). Θα πρέπει να σημειωθεί ότι ο κύβος είναι συμβολική ονομασία καθώς ένας κύβος δεδομένων μπορεί να έχει περισσότερες ή και λιγότερες από τρεις διαστάσεις. Ο κύβος δεδομένων πρόκειται ουσιαστικά για μια δομή παρόμοια με αυτή της διασταυρωμένης πινακοποίησης (cross table). [9, 14]

Δύο από τις βασικότερες ενέργειες που υποστηρίζουν τα συστήματα OLAP είναι αυτές του τεμαχισμού (slicing) και του κομματιάσματος (dicing). Πρόκειται για τις λειτουργίες που επιτρέπουν στον χρήστη να καθορίσει το υποσύνολο των κελιών του κύβου δεδομένων για το οποίο ενδιαφέρεται και θέλει να «εξορύξει» στατιστικά στοιχεία και πληροφορίες. Ο τεμαχισμός είναι ο καθορισμός μιας συγκεκριμένης τιμής για κάποια διάσταση, διαδικασία που οδηγεί σε έναν κύβο δεδομένων με μία διάσταση λιγότερη από τον αρχικό. Το κομμάτιασμα είναι ο προσδιορισμός ενός εύρους τιμών για κάποια διάσταση που οδηγεί αντίστοιχα σε έναν υποκύβο του αρχικού. Συνήθως για να καθοριστεί το υποσύνολο των κελιών που είναι προς μελέτη χρησιμοποιούνται συνδυαστικά οι δύο ενέργειες που αναφέρθηκαν προηγουμένως και για το λόγο αυτό η όλη διαδικασία του κατακερματισμού ενός κύβου δεδομένων είναι γνωστή ως slicing & dicing. [9, 14]



Σχήμα 2.3: Μετασχηματισμός του κύβου μετά από τεμαχισμό και κομμάτιασμα [13]

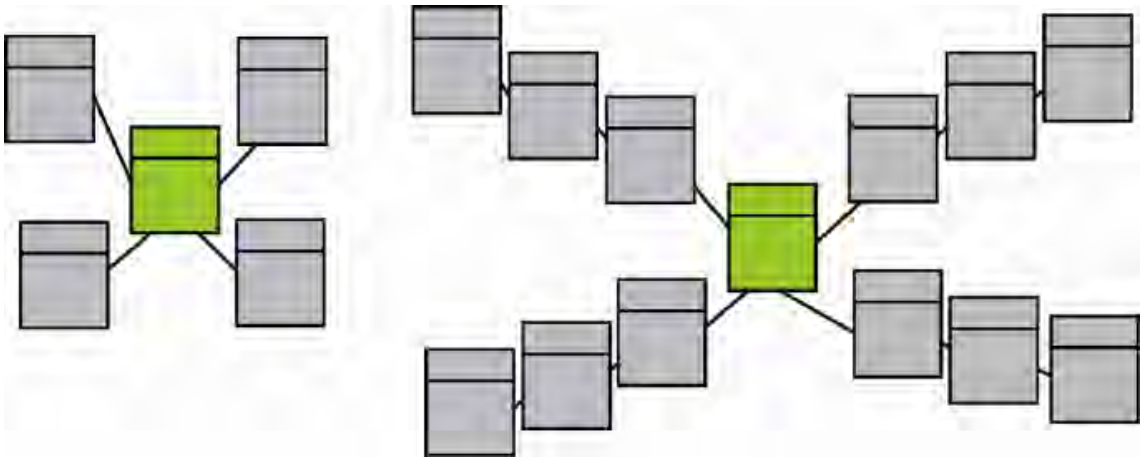
Οι διαστάσεις ενός κύβου μπορούν να περιέχουν διαφορετικά επίπεδα ιεραρχίας, παραδείγματος χάριν η διάσταση του χρόνου μπορεί να εκφραστεί σε χρονιές, μήνες και μέρες. Για τον λόγο αυτό μια ακόμα πολύ βασική ενέργεια των συστημάτων άμεσης αναλυτικής επεξεργασίας είναι αυτή της αναλυτικής καθόδου (Drill-down) κατά την οποία οι συναθροίσεις που έχουν υπολογιστεί σε ένα υψηλό επίπεδο, π.χ. για μια ολόκληρη χρονιά, μπορούν να αναλυθούν σε ένα χαμηλότερο επίπεδο δηλαδή για κάθε μήνα της συγκεκριμένης χρονιάς. Η αντίστροφη διαδικασία ονομάζεται συναθροιστική άνοδος (Roll-up). [9, 14]



Σχήμα 2.4: Μετασχηματισμός του κύβου με αναλυτική κάθοδο και συναθροιστική άνοδο [13]

Συνοψίζοντας τα συστήματα άμεσης αναλυτικής επεξεργασίας βασίζονται στην μοντελοποίηση των δεδομένων στη μορφή ενός πολυδιάστατου «κύβου» δεδομένων και στην συνάθροιση τους με κατάλληλο τρόπο και επιτυγχάνουν έτσι να φανερώσουν υποσύνολα που μπορεί να περιέχουν σημαντική πληροφορία. Όταν οι βάσεις δεδομένων στις οποίες βασίζονται τα συστήματα OLAP είναι σχεσιακές τότε ονομάζονται Relational OLAP ή ROLAP. [9, 14]

Τα δύο συνηθέστερα σχήματα βάσεων δεδομένων στα οποία βασίζονται τα συστήματα ROLAP είναι αυτά του αστέρα και της χιονονιφάδας. Το σχήμα αστέρα περιέχει έναν πίνακα γεγονότων (fact table) στο κέντρο ο οποίος συνδέεται με τους πίνακες των διαστάσεων, ενώ το σχήμα χιονονιφάδας ουσιαστικά αποτελεί μια επέκταση του σχήματος αστέρα καθώς σε αυτό ορισμένοι πίνακες διαστάσεων συνδέονται με άλλους μικρότερους πίνακες διαστάσεων που αντιστοιχούν στα διαφορετικά επίπεδα ιεραρχίας της διάστασης. Επίσης αξίζει να αναφερθεί και το σχήμα γαλαξία κατά το οποίο υπάρχουν περισσότεροι από ένας πίνακες γεγονότων οι οποίοι μπορεί να μοιράζονται ορισμένες διαστάσεις. [9, 14]



Σχήμα 2.5: Σχήμα αστέρα και σχήμα χιονονιφάδας [1]

2.3.1 Δημοφιλή Συστήματα Άμεσης Αναλυτικής Επεξεργασίας.

Κατά τα τελευταία χρόνια υπάρχει ραγδαία ανάπτυξη της Επιχειρησιακής Νοημοσύνης (Business Intelligence – BI), δηλαδή του συνόλου των τεχνικών που χρησιμοποιούνται από τις επιχειρήσεις και στοχεύουν στην υποστήριξη της λήψης αποφάσεων με βάση τα δεδομένα που συλλέγουν. Τέτοιες τεχνικές είναι η μηχανική μάθηση, η εξόρυξη δεδομένων, η συγκριτική προτυποποίηση, η άμεση αναλυτική επεξεργασία κ.α. Για τον λόγο αυτό υπάρχει διαθέσιμο στην αγορά μεγάλο πλήθος από συστήματα και εργαλεία που προσφέρουν τις λειτουργίες OLAP. Παρακάτω ακολουθούν ορισμένα από τα δημοφιλέστερα.

- **IBM Cognos**

  Αποτελεί μια διαδικτυακή πλατφόρμα επιχειρησιακής νοημοσύνης που περιέχει μεγάλο πλήθος εργαλείων έτσι ώστε να ανταποκρίνεται στις απαιτήσεις μιας επιχείρησης.

- **Micro Strategy**

  Ιδιαίτερα καινοτόμα πλατφόρμα που προσφέρει την δυνατότητα για πρόσβαση και αναλυτική επεξεργασία των δεδομένων της επιχείρησης από τους υπαλλήλους μιας επιχείρησης μέσα από κινητά τηλέφωνα και φορητές συσκευές με ασφαλή τρόπο.

- **Palo**

  Εξυπηρετητής πολυδιάστατης άμεσης αναλυτικής επεξεργασίας που χρησιμοποιεί λογισμικό λογιστικού φύλλου (π.χ. Excel) ως περιβάλλον χρήστη. Επιτρέπει σε πολλούς διαφορετικούς χρήστες να μοιράζονται μία κεντρικοποιημένη αποθήκη δεδομένων. Αποτελεί προϊόν της Jedox A.G.

- **Apache Kylin**

Λογισμικό ανοικτού κώδικα που έχει σχεδιαστεί για να προσφέρει διεπαφή SQL και δυνατότητες άμεσης αναλυτικής επεξεργασίας στο Apache Hadoop. Είναι ιδανικό για πολύ μεγάλες βάσεις δεδομένων που μπορεί να έχουν δισεκατομμύρια εγγραφές.

Τα παραπάνω συστήματα προορίζονται για βαριά επαγγελματική χρήση, για τις ανάγκες της παρούσας εργασίας απαιτείται ένα ελεύθερο και πολύ πιο ελαφρύ σύστημα άμεσης αναλυτικής επεξεργασίας καθώς ο όγκος των δεδομένων που θα χρησιμοποιηθούν θα είναι κατά πολύ μικρότερος από αυτά που συλλέγονται και χρησιμοποιούνται από μια επιχείρηση ενώ και οι απαιτήσεις δεν θα είναι τόσο αυστηρές καθώς τα συμπεράσματα που θα εξαχθούν από την ανάλυση των δεδομένων δεν θα διέπονται από υψηλή κρισιμότητα.

Για τον λόγο αυτό έχει επιλεχθεί το **Cubes**, το οποίο αποτελεί μέρος της **Data Brewery**, μιας σειράς από εργαλεία στην προγραμματιστική γλώσσα **Python** για την ανάλυση και την επεξεργασία δεδομένων. Το συγκεκριμένο εργαλείο μπορεί να χρησιμοποιηθεί ως διεπαφή μεταξύ μιας βάσης δεδομένων και μιας Python εφαρμογής για να της προσδώσει τις βασικότερες λειτουργίες ενός συστήματος OLAP. Το κεφάλαιο που ακολουθεί θα ασχοληθεί εκτενώς με το συγκεκριμένο εργαλείο ,τόσο από την άποψη της παρουσίασης της αρχιτεκτονικής του όσο και της περιγραφής του τρόπου με τον οποίο μπορεί να χρησιμοποιηθεί, λειτουργώντας έτσι ως ένας αναλυτικός οδηγός για το Cubes .

Κεφάλαιο 3

Παρουσίαση του Cubes

3.1 Εισαγωγή

Το Cubes αναπτύχθηκε στις αρχές της δεκαετίας από τον Steve Urbanek με στόχο να προσφέρει ένα ελαφρύ και εύχρηστο εργαλείο για αποθήκες δεδομένων που θα μπορεί να χρησιμοποιηθεί από εφαρμογές αναφοράς δεδομένων ή Python modules για την προετοιμασία και αναφορά πολυδιάστατων δεδομένων σε διαφορετικά επίπεδα, εστιάζοντας στην απλότητα και την ταχύτητα χωρίς να προσφέρει τον μεγάλο αριθμό δυνατοτήτων άλλων «βαρέων» προγραμμάτων για αποθήκες δεδομένων. Πρόκειται ουσιαστικά για ένα εργαλείο υλοποιημένο στην προγραμματιστική γλώσσα Python σε συνδυασμό με έναν ενσωματωμένο HTTP Server για την επίτευξη της λειτουργικότητας OLAP. Επίσης το Cubes μπορεί να υποστηρίξει πληθώρα διαφορετικών βάσεων δεδομένων, τόσο σχεσιακές όσο και μη-σχεσιακές, και καθίσταται έτσι ιδιαίτερα προσαρμόσιμο και λειτουργικό ανεξάρτητα με την μορφή που έχουν τα δεδομένα. [12]

Η αρχιτεκτονική του Cubes συνίσταται σε τέσσερα «κομμάτια» ο συνδυασμός των οποίων συνθέτει τον εργασιακό χώρο (workspace) που μας παρέχει την επιθυμητή λειτουργικότητα, αυτά είναι το λογικό μοντέλο, ο περιηγητής συναθροίσεων, ο εξυπηρετητής και τα εσωτερικά μέρη. Στην συνέχεια θα παρουσιαστούν εκτενώς τα τέσσερα προαναφερθέντα μέρη και όλες οι βασικές έννοιες που απαιτούνται για την καλύτερη κατανόηση της δομής και της λειτουργίας του. [12]

Για την εγκατάσταση του Cubes χρειάζεται ο pip installer, εφόσον αυτός βρίσκεται εγκατεστημένος στον υπολογιστή στον οποίο θα χρησιμοποιηθεί το Cubes μένει να δοθεί στην γραμμή εντολών η ακόλουθη εντολή:

pip install cubes[all]

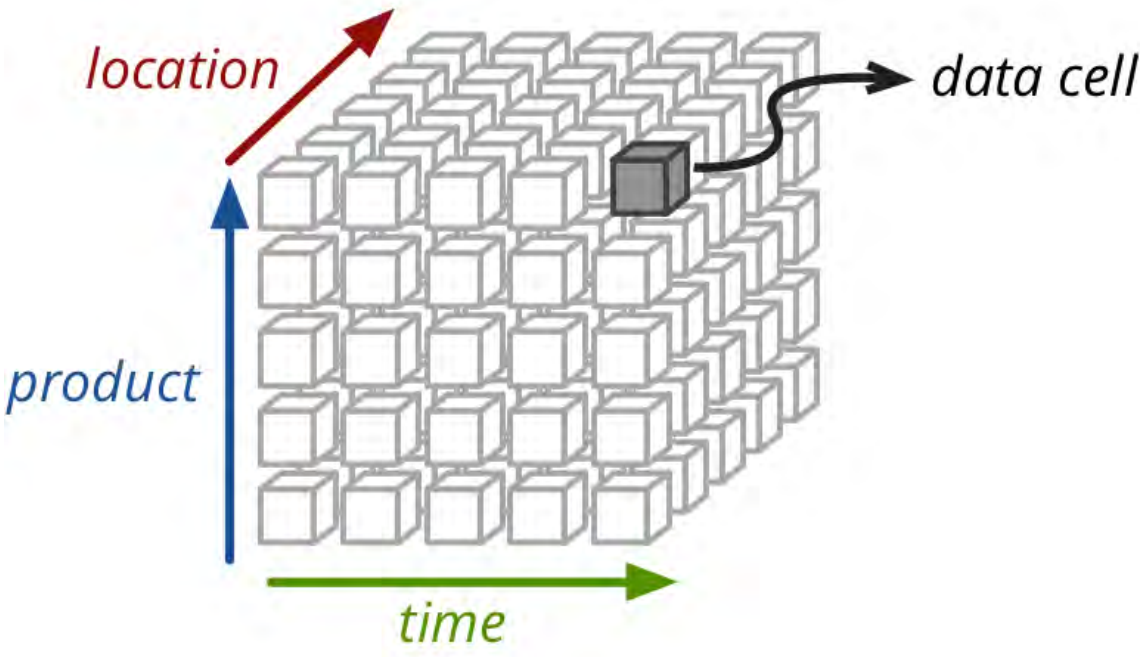
Ο κώδικας πηγής του εργαλείου Cubes μπορεί να βρεθεί στον ακόλουθο σύνδεσμο:

https://github.com/DataBrewery/cubes

3.2 Λογικό Μοντέλο (Model)

Το μοντέλο είναι αυτό που επιτυγχάνει να προσδώσει στα δεδομένα την οπτική την οποία ενδιαφέρει τον χρήστη ανεξαρτήτως της φυσικής τους υλοποίησης. Σκοπός είναι δηλαδή να μοντελοποιηθούν με τέτοιο τρόπο ώστε να καθίσταται ευνόητο το τι μπορεί να μετρηθεί σε αυτά και οι παράμετροι υπό τις οποίες μπορούν να κατηγοριοποιηθούν, κατά αυτόν τον τρόπο επιτυγχάνεται να «κρυφτεί» το πώς είναι κατανεμημένα στην βάση δεδομένων που μπορεί να είναι αχανής και ιδιαίτερα πολύπλοκη. Αρχικά θα γίνει μια επανάληψη στις βασικές έννοιες ενός OLAP Cube ενώ στην συνέχει θα εξηγηθεί το πώς δημιουργείται το μοντέλο που εφαρμόζεται από το Cubes πάνω στα δεδομένα. [12]

- Fact: Το μικρότερο κομμάτι ενός κύβου ονομάζεται Fact και πρόκειται για το αντικείμενο το οποίο θέλουμε να μετρηθεί (π.χ. ένα συμβόλαιο, μια κλήση)
- Measure: Τα μεγέθη τα οποία μετριούνται για κάποιο Fact, π.χ. για μια κλήση μπορεί να μετρηθεί η διάρκεια και το κόστος
- Dimension: Οι διαστάσεις προσφέρουν το εννοιολογικό πλαίσιο για τα Facts όπως π.χ. η ημερομηνία ή το είδος. Κάθε διάσταση μπορεί να έχει πολλά επίπεδα ιεραρχίας
- Cube: Το σύνολο όλων των δεδομένων που αφορούν μια συγκεκριμένη ομάδα από Facts. Τα δεδομένα μοντελοποιούνται ως μια συλλογή από Cubes με πολλαπλές διαστάσεις (μπορούν να είναι περισσότερες ή και λιγότερες από τρεις)



Σχήμα 3.1: Παράδειγμα ενός Cube [12]

Για να δημιουργηθεί ένα μοντέλο θα πρέπει να οριστεί ένα αρχείο JSON το οποίο θα περιγράφει τις διαστάσεις και τους κύβους που το συνιστούν. Για να γίνει περισσότερο κατανοητό θα χρησιμοποιηθεί ένα παράδειγμα. Υποθέτοντας ότι έχουμε δεδομένα από πωλήσεις προϊόντων και θέλουμε να ορίσουμε έναν κύβο που θα αντιπροσωπεύει το σύνολο όλων των μοναδικών πωλήσεων που υπάρχουν στην βάση δεδομένων διαταγμένες κατά τρεις διαστάσεις, το είδος του προϊόντος, τον τρόπο πληρωμής και το μέρος που έλαβε χώρα η συναλλαγή. Περαιτέρω για την τρίτη διάσταση θα υπάρχουν τρία επίπεδα ιεραρχίας, η χώρα, η περιφέρεια και η πόλη.

Ακολουθούν τα σημαντικότερα μεταδεδομένα που πρέπει να οριστούν ξεκινώντας από αυτά που αφορούν το μοντέλο και στην συνέχεια αυτά των διαστάσεων και των κύβων. Αρχικά για το μοντέλο υπάρχουν τα εξής μεταδεδομένα:

name	Όνομα του μοντέλου
label	Όνομα του μοντέλου όπως φαίνεται στον χρήστη
description	Περιγραφή
locale	Χρησιμοποιείτε σε localizable μοντέλα
cubes	Μεταδεδομένα των κύβων του μοντέλου
dimensions	Μεταδεδομένα των διαστάσεων του μοντέλου
public_dimensions	Διαστάσεις που μπορούν να χρησιμοποιηθούν από κύβους άλλων μοντέλων. Εάν δεν οριστούν όλες οι διαστάσεις θεωρούνται δημόσιες.
store	Όνομα του datastore όπου βρίσκονται οι κύβοι του μοντέλου. Εάν δεν οριστεί επιλέγεται το προεπιλεγμένο datastore που έχει οριστεί στον workspace
mappings	Αντιστοιχίσεις με τα φυσικά δεδομένα. Κληροδοτούνται σε όλους του κύβους του μοντέλου
joins	Προσδιορισμός των συνδέσεων (joins). Κληροδοτούνται σε όλους του κύβους του μοντέλου
browser_options	Επιλογές για τον περιηγητή. Προστίθενται στις πιο συγκεκριμένες επιλογές που ορίζονται στους κύβους

Πίνακας 3.1: Μεταδεδομένα Μοντέλου [12]

Τα εφτά πρώτα χαρακτηριστικά συνιστούν το λογικό κομμάτι της περιγραφής του μοντέλου ενώ τα υπόλοιπα τέσσερα το φυσικό κομμάτι. Στον παρακάτω κώδικα φαίνεται πως ορίζετε το μοντέλο για το παράδειγμα με τις πωλήσεις που προαναφέρθηκε.

```
{
  ”name”: ”sales_model”,
  ”label”: ”Model of Sales”,
  ”cubes”: [...],
  ”dimensions”: [...],
  ”mappings”: [...]
}
```

Τα σημαντικότερα μεταδεδομένα που αφορούν την περιγραφή των διαστάσεων είναι τα ακόλουθα:

name	Όνομα της διάστασης
label	Όνομα της διάστασης όπως φαίνεται στον χρήστη
description	Περιγραφή
info	Πληροφορίες
levels	Τα επίπεδα που περιλαμβάνει η διάσταση (εάν είναι περισσότερα του ενός)
hierarchies	Ο τρόπος ή οι τρόποι που ιεραρχούνται τα επίπεδα
default_hierarchy_name	Προκαθορισμένη ιεραρχία

Πίνακας 3.2: Μεταδεδομένα Διάστασης [12]

Παρακάτω ορίζονται οι διαστάσεις για το παράδειγμα των πωλήσεων. Για την πρώτη διάσταση, αυτή της ”geography”, έχουν οριστεί τρία επίπεδα ενώ οι άλλες δύο διαστάσεις έχουν ένα μοναδικό επίπεδο. Να σημειωθεί ότι δεν έχει οριστεί κάποια ιεραρχία για την ”geography” που σημαίνει ότι θα ισχύει η προκαθορισμένη ιεραρχία η οποία θα περιλαμβάνει όλα τα επίπεδα με την σειρά που τα έχουμε ορίσει δηλαδή ”country” -> ”state” -> ”city”.

```
”dimensions”: [
{
  ”name”:”geography”,
  ”levels”: [
  {
    ”name”:”country”,
    ”label”:”Country”
  },
  {
    ”name”:”state”,
    ”label”:”State/Region”
  },
  {
    ”name”:”city”,
    ”label”:”City”
  } ]
},
{
  ”name”:”payment”,
  ”label”:”Payment Type”
},
{
  ”name”:”product”,
  ”label”:”Product Type”
} ],
```

Τα μεταδεδομένα που αφορούν την περιγραφή των κύβων είναι τα ακόλουθα:

name	Όνομα του κύβου
label	Όνομα του κύβου όπως φαίνεται στον χρήστη
description	Περιγραφή του κύβου
info	Ειδικές Πληροφορίες. Δεν χρησιμοποιούνται από το framework
dimensions	Διαστάσεις του κύβου
measures	Λίστα με τα μέτρα του κύβου
aggregates	Λίστα των συναθροίσεων που προκύπτουν από την εφαρμογή συναρτήσεων πάνω στα μέτρα
details	Λίστα χαρακτηριστικών που δεν χρησιμοποιούνται στις συναθροίσεις αλλά θέλουμε να εμφανίζονται κατά την επίδειξη των facts
joins	Προσδιορισμός των συνδέσεων.
mappings	Αντιστοιχίσεις με τα φυσικά δεδομένα.
browser_options	Επιλογές για τον περιηγητή.
store	Χρησιμοποιείται όταν ο κύβος βρίσκεται σε διαφορετικό store από το προκαθορισμένο

Πίνακας 3.3: Μεταδεδομένα Κύβου [12]

Με την χρήση των παραπάνω μεταδεδομένων έχει οριστεί ένας κύβος για τις πωλήσεις σύμφωνα με το παράδειγμα που έχει προαναφερθεί. Όπως φαίνεται παρακάτω έχουν δηλωθεί ως διαστάσεις του κύβου οι τρεις (”geography”, ”payment”, ”product”) που ορίστηκαν προηγουμένως. Στο σημείο αυτό να σημειωθεί πως σε ένα πιο σύνθετο μοντέλο το οποίο θα είχε δύο κύβους, π.χ. έναν για τις πωλήσεις της εταιρίας και έναν για τις αγορές της από προμηθευτές, θα μπορούσαμε να έχουμε κάποιες διαστάσεις κοινές στους δύο κύβους και κάποιες άλλες αποκλειστικά για τον καθένα. Πίσω στο παράδειγμα των πωλήσεων, το μοναδικό μέτρο (measure) είναι η τιμή για κάθε αγοραπωλησία ”price”. Έχουν οριστεί δύο συναθροίσεις (aggregates), η πρώτη (amount) χρησιμοποιεί την συνάρτηση sum πάνω στο μέτρο ”price” για να αθροίσει τις τιμές για όλες τις πωλήσεις του κύβου ενώ η δεύτερη (records) δεν χρησιμοποιεί κάποιο μέτρο αλλά είναι ένας απλός μετρητής των πωλήσεων του κύβου με την συνάρτηση count. Με τον τρόπο αυτό έχει επιτευχθεί να μπορούμε να λάβουμε ως αποτέλεσμα το συνολικό κόστος και το πλήθος των πωλήσεων για οποιαδήποτε υποσύνολο του κύβου επιθυμούμε βάσει των τριών διαστάσεων π.χ. πόσες πωλήσεις του προϊόντος Α έγιναν σε μια συγκεκριμένη χώρα και ποιο ήταν το συνολικό κόστος.

Τέλος, με την υπόθεση ότι στο παράδειγμα αυτό τα ονόματα που έχουν δοθεί στις διαστάσεις και τα μέτρα του κύβου ταυτίζονται με τα ονόματα των αντίστοιχων στηλών του πίνακα γεγονότων της βάσης δεδομένων, οι μόνες αντιστοιχίσεις με τα φυσικά δεδομένα που θα πρέπει να οριστούν ρητά είναι αυτές των επιπέδων της διάστασης ”geography” όπως φαίνονται παρακάτω, με την υπόθεση

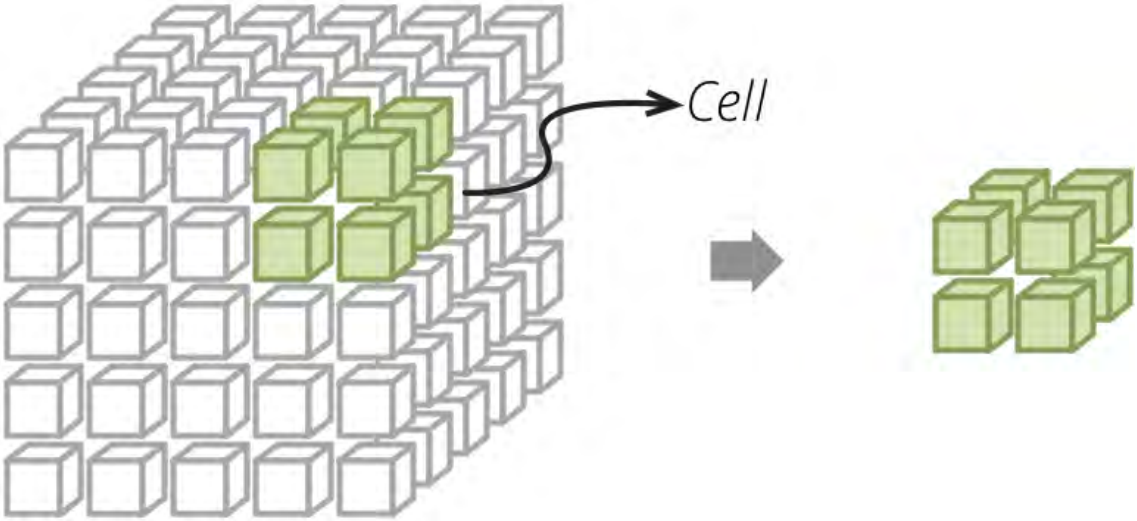
πως τα ονόματα των στηλών στην βάση δεδομένων είναι ”country”, ”state” και ”city”. Περισσότερα για το τελευταίο θα αναλυθούν και θα γίνουν περισσότερο κατανοητά στην υποενότητα που αφορά την λειτουργία του SQL Backend.

```
”cubes”: [ {
  ”name”:”sales”,
  ”dimensions”: [”geography”, ”payment”, ”product”],
  ”measures”:[”name”:”price”,”label”:”Price”],
  ”aggregates”:[ {
    ”name”:”amount”,
    ”function”:”sum”,
    ”measure”:”price”,
  },
  {
    ”name”:”records”,
    ”function”:”count”,
  } ],
  ”mappings”: {
    ”geography.country”:”country”,
    ”geography.state”:”state”,
    ”geography.city”:”city”
  }
} ]
```

3.3 Περιηγητής Συναθροίσεων (Aggregation Browser)

Ο περιηγητής ευθύνεται για την αναλυτική επεξεργασία των δεδομένων. Έπειτα από την εφαρμογή του μοντέλου πάνω στα φυσικά δεδομένα ο περιηγητής είναι αυτός που επιτρέπει την «περιήγηση» μέσα σε αυτά και υπολογίζει τις κατάλληλες συναθροίσεις έτσι ώστε να καθίσταται δυνατή η εξαγωγή συμπερασμάτων από τον χρήστη. Η βασικότερη έννοια του περιηγητή συναθροίσεων είναι το κελί (cell), δηλαδή το κομμάτι του κύβου που αντιστοιχεί στο υποσύνολο των δεδομένων που απασχολεί τον χρήστη την κάθε δεδομένη φορά. [12]

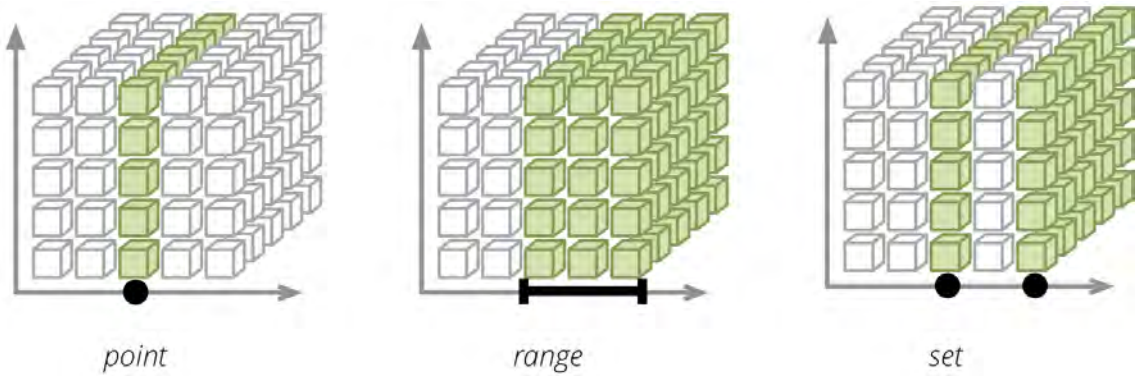
Η διαδικασία κατά την οποία διαιρείτε ο κύβος σε μικρότερα κομμάτια με σκοπό να απομείνει το υποσύνολο των δεδομένων που απασχολεί τον χρήστη, δηλαδή ο καθορισμός του κελιού ονομάζεται slicing & dicing. [12] Για να επιτευχθεί αυτό εφαρμόζονται ορισμένες τομές (cuts) στον αρχικό κύβο, το Cubes υποστηρίζει τρεις διαφορετικούς τύπους τομών :



Σχήμα 3.2: Προσδιορισμός του κελιού από τον κύβο [12]

- Point Cut: Καθορίζει μια συγκεκριμένη τιμή σε κάποια διάσταση
- Range Cut: Καθορίζει ένα εύρος τιμών σε κάποια διατεταγμένη διάσταση
- Set Cut: Καθορίζει μια συλλογή συγκεκριμένων τιμών σε κάποια διάσταση

Ο κατάλληλος συνδυασμός των παραπάνω τομών μπορεί να προσδιορίσει οποιοδήποτε κομμάτι του κύβου επιθυμεί ο χρήστης.



Σχήμα 3.3: Οι τρεις τύποι τομών [12]

Παρακάτω φαίνεται πως μπορούν να υλοποιηθούν τα παραπάνω σε Python για να καθοριστεί ένα κελί στο παράδειγμα των πωλήσεων που θα περιλαμβάνει όλες τις αγοροπωλησίες που έγιναν με Visa ως τρόπο πληρωμής στην Ελλάδα και την Ισπανία. Για να επιτευχθεί αυτό απαιτούνται δύο τομές, μία point cut στην διάσταση ”payment” και μία set cut στην διάσταση ”geography”.

```
cut = [
  PointCut("payment", ["Visa"]),
  SetCut("geography",[["Greece"],["Spain"]],hierarchy="default")
]

cell = Cell(browser.cube,cut)
```

Εφόσον καθοριστεί το κελί καθίσταται δυνατή η λειτουργία του υπολογισμού των συναθροίσεων με την χρήση της ρουτίνας "aggregate" του περιηγητή:

aggregate(cell=None, aggregates=None, drilldown=None, split=None, order=None, page=None, page_size=None, **options)

Οι σημαντικότερες είσοδοι της εν λόγω συνάρτησης με τις οποίες μπορεί να επιτευχθεί η επιθυμητή λειτουργικότητα είναι οι τρεις πρώτες. Εάν δεν δοθεί ως είσοδος το όνομα του κελιού που καθορίστηκε προηγουμένως τότε το aggregation θα γίνει για ολόκληρο τον κύβο, επίσης μπορούν να δηλωθούν ρητά οι συναθροίσεις του μοντέλου που ενδιαφέρουν τον χρήστη στην κάθε περίπτωση ενώ τέλος το τρίτο input δηλαδή η δυνατότητα εκτέλεσης αναλυτικής καθόδου θα επεξηγηθεί εκτενώς στην παράγραφο που ακολουθεί. Επίσης μια ακόμα ρουτίνα που αξίζει να αναφερθεί είναι η **facts(cell=None)** η οποία επιστρέφει όλα τα facts που συνθέτουν ένα κελί [12].

```
## Όλος ο κύβος και όλες οι συναθροίσεις
result = browser.aggregate()

## Το cell που καθορίσαμε παραπάνω και όλες οι συναθροίσεις
result = browser.aggregate(cell)

## Το cell που καθορίσαμε παραπάνω και μόνο το amount
result = browser.aggregate(cell, aggregates=["amount"])

## Όλα τα facts που συνθέτουν το cell
result = browser.facts(cell)
```

Η αναλυτική κάθοδος (drilldown) προσφέρει στον αναλυτή περισσότερες λεπτομέρειες καθώς ομαδοποιεί το αποτέλεσμα κατά κάποια διάσταση, για παράδειγμα μπορεί να εμφανίσει τα αποτελέσματα ανά είδος προϊόντος ή ανά πόλη. Εάν κάποια διάσταση έχει πολλαπλά επίπεδα ιεραρχίας, όπως η διάσταση "geography" η οποία έχει τρία επίπεδα (χώρα, περιφέρεια, πόλη), τότε η αναλυτική κάθοδος θα γίνει για το αμέσως επόμενο επίπεδο ιεραρχίας από αυτό στο οποίο έχει γίνει η βαθύτερη τομή. Δηλαδή στο προηγούμενο παράδειγμα εάν έχει καθοριστεί ένα κελί για τις αγοροπωλησίες που έχουν γίνει στην Ελλάδα εάν ζητηθεί αναλυτική κάθοδος κατά "geography" αυτό θα γίνει στο επίπεδο περιφέρειας καθώς η βαθύτερη τομή στην διάσταση αυτή έχει γίνει σε επίπεδο χώρας. Για να εκτελεστεί η αναλυτική κάθοδος και να γίνει η κατηγοριοποίηση σε ένα βαθύτερο επίπεδο της διάστασης θα πρέπει να δηλωθεί ρητά όπως φαίνεται στον παρακάτω κώδικα. Συνεπώς ο καθορισμός της αναλυτικής καθόδου γίνεται με δύο τρόπος είτε με το όνομα της διάστασης

όταν εκτελείται για το αμέσως επόμενο επίπεδο είτε με την τριάδα (διάσταση, ιεραρχία, επίπεδο) όταν εκτελείται για κάποιο βαθύτερο επίπεδο. [12]

```
## Drill-down ανά είδος προϊόντος
result = browser.aggregate(cell,drilldown=[”product”])
for record in result:
  print(record)

## Drill-down ανά πόλη
result = browser.aggregate(cell,drilldown=[(”geography”,
”default”, ”city”)])
for record in result:
  print(record)
```

Τέλος απαραίτητο κομμάτι για την υλοποίηση του περιηγητή συναθροίσεων είναι να οριστεί ο χώρος εργασίας (Workspace), δηλαδή να οριστεί ποιο λογικό μοντέλο θα χρησιμοποιηθεί πάνω σε ποια δεδομένα. [12] Αυτό σε προγραμματιστικό περιβάλλον Python, με την υπόθεση πως χρησιμοποιούμε SQLite , γίνετε ως ακολούθως:

```
## WORKSPACE

ws = Workspace()
ws.register_default_store(”sql”, url=”sqlite:///data.sqlite”)
ws.import_model(”sales_model.json”)
```

3.4 Εξυπηρετητής (Slicer Server)

Για να είναι δυνατή η χρήση του framework από χρήστες και εφαρμογές χωρίς να απαιτείται η γνώση της γλώσσας Python, υπάρχει ενσωματωμένος ένας απλός HTTP OLAP Server που ονομάζεται Slicer. Ο εξυπηρετητής αυτός καλύπτει τις περισσότερες από τις δυνατότητες του περιηγητή συναθροίσεων (slicing & dicing, drilldowns, aggregations, κ.α.) προσφέροντας έτσι ένα επίπεδο αφαίρεσης στην εφαρμογή. Κατά αυτόν τον τρόπο ένας χρήστης/εφαρμογή μπορεί να κάνει χρήση της λειτουργικότητας OLAP του Cubes υποβάλλοντας ένα απλό HTTP Request. Η απάντηση που θα λάβει θα είναι σε μορφότυπο JSON το οποίο είναι ιδιαίτερα διαχειρίσιμο και μπορεί να αξιοποιηθεί για παράδειγμα από μια εφαρμογή για την οπτικοποίηση των αποτελεσμάτων. [12]

Ακολουθούν ορισμένα από τα σημαντικότερα HTTP Requests:

- **GET /info** : Επιστρέφει πληροφορίες για τον server
- **GET /cubes** : Επιστρέφει τους κύβου που υπάρχουν στο μοντέλο
- **GET /cube/<name>/model** : Επιστρέφει πληροφορίες για τον συγκεκριμένο κύβο (λίστα από συναθροίσεις, μέτρα, κτλ)

- **GET /cube/<name>/aggregate** : Επιστρέφει τα αποτελέσματα του aggregation
- **GET /cube/<name>/members/<dim>** : Επιστρέφει τα μέλη μια διάστασης
- **GET /cube/<name>/facts** : Επιστρέφει τα facts που ανήκουν στον κύβο

Μένει τώρα να αποσαφηνιστεί πως ενσωματώνονται στις παραπάνω αιτήσεις (στο aggregate και το facts συγκεκριμένα) οι απαραίτητες παράμετροι για να εκτελεστούν τομές και αναλυτικές κάθοδοι. Αρχικά για να προστεθούν παράμετροι σε μια αίτηση θα πρέπει αυτό να ακολουθείται από ένα αγγλικό ερωτηματικό (?) και στην συνέχεια να ακολουθούν οι επιθυμητές παράμετροι. Παρακάτω φαίνεται πως μπορούν να οριστούν διάφορες τομές και να εκτελεστούν αναλυτικές κάθοδοι που καλύπτουν τις περισσότερες περιπτώσεις.

```
## Πωλήσεις στην Ελλάδα
GET /cube/sales/aggregate?cut=geography:Greece

## Πωλήσεις στην Αθήνα
GET /cube/sales/aggregate?cut=geography:Greece,Attiki,Athens

## Πωλήσεις στην Τουρκία με Visa
GET /cube/sales/aggregate?cut=geography:Turkey|payment:Visa

## Πωλήσεις στην Ελλάδα και την Ισπανία (Set Cut)
GET /cube/sales/aggregate?cut=geography:Greece;Spain

## Πωλήσεις από το 2013 έως το 2017 (Range Cut)
GET /cube/sales/aggregate?cut=date:2013-2017

## Πωλήσεις από την αρχή των δεδομένων έως το 2017 (Open Range
Cut)
GET /cube/sales/aggregate?cut=date:-2017

## Πωλήσεις ανά χώρα
GET /cube/sales/aggregate?drilldown=geography

## Πωλήσεις ανά πόλη
GET /cube/sales/aggregate?drilldown=geography:city

## Πωλήσεις με Visa ανά χώρα
GET /cube/sales/aggregate?drilldown=geography&cut=payment:Visa
```

Για να χρησιμοποιηθεί ο εξυπηρετητής Slicer και να αξιοποιηθούν όλες οι δυνατότητες που αναφέρθηκαν παραπάνω θα πρέπει να δημιουργηθεί ένα αρχείο .ini για την κατάλληλη αρχικοποίηση του. Το αρχείο αυτό θα πρέπει να «δείχνει» στον εξυπηρετητή που είναι τα δεδομένα και ποιο είναι

το αρχείο στο οποίο έχει οριστεί το μοντέλο, όπως δηλαδή ορίστηκε ο εργασιακός χώρος προηγουμένως για τον περιηγητή συναθροίσεων. Εάν για παράδειγμα τα δεδομένα είναι αποθηκευμένα στο SQLite η αρχικοποίηση θα γίνει ως εξής:

```
[store]
type: sql
url: sqlite:///data.sqlite

[models]
model: sales_model.json
```

Τέλος για να «τρέξει» ο εξυπηρετητής Slicer αρκεί να δοθεί η εντολή:

slicer serve slicer.ini

3.5 Εσωτερικά Μέρη (Backends)

Το εσωτερικό μέρος καλύπτει τις μη-ορατές από τον χρήστη διεργασίες και είναι υπεύθυνο για όλες τις ενέργειες που διεξάγονται πάνω στα «φυσικά» δεδομένα. Ουσιαστικά αποτελεί το χαμηλότερο επίπεδο του framework και είναι αυτό στο οποίο βασίζεται ο περιηγητής συναθροίσεων για την λειτουργία του. Το Cubes περιλαμβάνει πληθώρα διαφορετικών εσωτερικών μερών έτσι ώστε να μπορεί να υποστηρίξει πολλών τύπων data stores, περαιτέρω είναι δυνατή και η ταυτόχρονη χρήση διαφορετικών εσωτερικών μερών.[12] Συγκεκριμένα περιλαμβάνονται:

1. SQL Backend
2. MongoDB Backend
3. Mixpanel Backend
4. Slicer Server
5. Google Analytics Backend

Σχήμα 3.4: Διαφορετικά εσωτερικά μέρη που υποστηρίζονται από το Cubes [11]

Στην πιο πρόσφατη έκδοση του Cubes υποστηρίζονται περισσότερο το εσωτερικό μέρος SQL και αυτό του εξυπηρετητή Slicer τα οποία θα χρησιμοποιηθούν και στην συγκεκριμένη εργασία. Παρακάτω θα παρουσιαστεί αναλυτικά το εσωτερικό μέρος SQL.

Το SQL Backend είναι υλοποιημένο με τη χρήση του SQLAlchemy το οποίο αποτελεί τον πλέον διαδεδομένο Object-Relational Mapper (ORM) για την γλώσσα Python και υποστηρίζει πληθώρα διαλέκτων SQL όπως τις MySQL, Oracle, PostgreSQL, SQLite, Microsoft SQL Server και άλλες. Το SQLAlchemy έχει το ρόλο ενός διερμηνέα μεταξύ ενός περιβάλλοντος Python και του συστήματος διαχείρισης βάσεων δεδομένων (DBMS) που χρησιμοποιείται. Για την επίτευξη της αναλυτικής επεξεργασίας χρησιμοποιούνται απλές συναρτήσεις όπως οι sum, count, min, max, avg, stddev και variance οι οποίες εκτελούνται εν τέλει από το σύστημα διαχείρισης βάσεων δεδομένων. [12]

Όσον αφορά τις βάσεις δεδομένων που υποστηρίζει το εσωτερικό μέρος SQL του Cubes αυτές θα πρέπει να είναι σε σχήμα αστέρα ή χιονονιφάδας. Επίσης μπορεί προφανώς να υποστηρίξει και οποιοδήποτε μη κανονικοποιημένο πίνακα (denormalized table), δηλαδή μια βάση δεδομένων που αποτελείται από έναν μόνο πίνακα. Συνεπώς σε περίπτωση που τα δεδομένα δεν βρίσκονται σε κάποια από τις παραπάνω μορφές θα πρέπει να επεξεργασθούν κατάλληλα και να μετασχηματιστούν στην μη κανονικοποιημένη μορφή για να είναι δυνατή η χρήση του Cubes πάνω σε αυτά. [12]

Σχήμα 3.5: Σχήματα της βάσης δεδομένων που υποστηρίζονται από το Cubes [11]

Στο σημείο αυτό θα πρέπει να εξηγηθεί πώς το λογικό μοντέλο που έχει δημιουργηθεί και συνεπώς οι κύβοι, οι διαστάσεις και τα μέτρα που αυτό περιλαμβάνει, θα αντιστοιχιστεί πάνω στις στήλες των πινάκων της σχεσιακής βάσης δεδομένων, δηλαδή συνοπτικά πως γίνεται η σύνδεση λογικού και φυσικού επιπέδου. Αυτό επιτυγχάνεται συνδέοντας τους κατάλληλους πίνακες της βάσης δεδομένων και αντιστοιχώντας την επιθυμητή στήλη σε κάθε χαρακτηριστικό του μοντέλου. Κατά την προκαθορισμένη λειτουργία ένας κύβος «ψάχνει» στην βάση δεδομένων κάποιον πίνακα με το ίδιο όνομα, π.χ. ένας κύβος με όνομα ”sales” θα αντιστοιχιστεί με τον πίνακα γεγονότων με όνομα ”sales” ενώ για τα διάφορα χαρακτηριστικά των διαστάσεων αναμένεται ένας πίνακας με το όνομα της διάστασης και μια στήλη με το όνομα του χαρακτηριστικού, στην περίπτωση μιας

διάστασης με ένα μόνο επίπεδο που συνεπώς δεν χρειάζεται να υπάρχει διαφορετικός πίνακας στην βάση δεδομένων τότε η διάσταση αυτή αντιστοιχίζεται απ' ευθείας με την κατάλληλη στήλη του πίνακα γεγονότων.[12] Για να γίνει το παραπάνω περισσότερο κατανοητό ένα χαρακτηριστικό ”geography.country” θα αντιστοιχηθεί με την στήλη ”country” του πίνακα ”geography” ενώ το χαρακτηριστικό ”product” θα αντιστοιχηθεί με την στήλη ”product” του πίνακα ”sales” δηλαδή του πίνακα γεγονότων της βάσης δεδομένων.

Όπως γίνεται κατανοητό για να λειτουργήσει σωστά η προκαθορισμένη σύνδεση του μοντέλου στην βάση δεδομένων θα πρέπει αυτή να βρίσκεται σε σχήμα αστέρα και τα ονόματα των πινάκων και των στηλών να είναι ίδια με αυτά του λογικού μοντέλου. Σε διαφορετική περίπτωση θα πρέπει να δηλωθούν ρητά οι αντιστοιχίσεις (mappings) κατά τον ορισμό του μοντέλου αλλά και οι απαραίτητες συνδέσεις (joins). Παρακάτω φαίνεται πως μπορούν να δηλωθούν οι αντιστοιχίσεις και οι συνδέσεις στο μοντέλο έτσι ώστε να επιτυγχάνετε η επιθυμητή λειτουργικότητα από το εσωτερικό μέρος.

```
”mappings”: {
  ”logical_name”: ”physical name”
}

”joins” = [ {
  ”master”: ”fact_key”,
  ”detail”: ”dim_key”,
  ”method”:”match”
} ]


## method:
## match - INNER JOIN
## master - LEFT OUTER JOIN
## detail - RIGHT OUTER JOIN
```

Κεφάλαιο 4

Χρήση του Cubes για την αναλυτική επεξεργασία ποδοσφαιρικών δεδομένων

4.1 Εισαγωγή

Στο παρόν κεφάλαιο θα στηθεί μια εφαρμογή σε προγραμματιστικό περιβάλλον Python η οποία θα χρησιμοποιεί το εργαλείο Cubes για την άμεση αναλυτική επεξεργασία ενός dataset με δεδομένα που αφορούν το Παγκόσμιο Κύπελλο του 2018. Αρχικά θα παρουσιαστεί το σετ δεδομένων, στην συνέχεια θα δημιουργηθεί το λογικό μοντέλο που θα χρησιμοποιηθεί από το Cubes και τέλος θα παρουσιαστεί η εφαρμογή και θα παρατεθούν αποσπάσματα κώδικα σε γλώσσα Python.

4.2 Παρουσίαση Σετ Δεδομένων

Για την παρούσα εργασία έχει δημιουργηθεί μια βάση δεδομένων που αφορά το Παγκόσμιο Κύπελλο Ποδοσφαίρου του 2018 που διεξήχθη στα γήπεδα της Ρωσίας. Συγκεκριμένα περιλαμβάνει στοιχεία και γενικές πληροφορίες για όλους τους ποδοσφαιριστές που στελέχωσαν τις 32 ομάδες που συμμετείχαν στην διοργάνωση. Πρόκειται δηλαδή για μια βάση δεδομένων 736 εγγραφών για τις οποίες περιλαμβάνονται 14 γνωρίσματα τα οποία παρουσιάζονται στον πίνακα 4.1

Όσον αφορά τον τρόπο συλλογής των δεδομένων θα πρέπει να αναφερθούν οι παραδοχές που λήφθηκαν κατά την διαδικασία. Καταρχάς οι σύλλογοι των ποδοσφαιριστών είναι αυτοί με τους οποίους αγωνίζονταν κατά την ολοκλήρωση της ποδοσφαιρικής περιόδου 2017-2018 ενώ η ηλικία τους είναι σύμφωνα με την ημερομηνία έναρξης της διοργάνωσης. Περαιτέρω στα λεπτά συμμετοχής δεν προσμετρούνται αυτά των καθυστερήσεων των αγώνων, συνεπώς ένας ποδοσφαιριστής που έχει αγωνιστεί δύο φορές ως αλλαγή στις καθυστερήσεις του αγώνα θα έχει δύο συμμετοχές αλλά μηδέν λεπτά συμμετοχής. Για την μέτρηση των κόκκινων καρτών προσμετρούνται τόσο αυτές που δέχθηκε ο ποδοσφαιριστής απ' ευθείας όσο και αυτές που δέχθηκε ως δεύτερη κίτρινη κάρτα. Για την μέτρηση των τελικών πασών, το οποίο είναι ένα μέτρο που δεν μπορεί εκ φύσεως να μετρηθεί απολύτως αντικειμενικά, έχει χρησιμοποιηθεί η μέτρηση από την έγκριτη ιστοσελίδα ποδοσφαιρικών δεδομένων transfermarkt. [10] Τέλος οι συνολικές συμμετοχές και τα συνολικά

τέρματα των ποδοσφαιριστών με τις εθνικές τους ομάδες είναι σύμφωνα με την ημερομηνία έναρξης της διοργάνωσης και συνεπώς οι συμμετοχές και τα γκολ σε αυτήν δεν προσμετρούνται στα γνωρίσματα αυτά.

Γνώρισμα	Περιγραφή
NAME	Όνομα Ποδοσφαιριστή
NAT_TEAM	Εθνική ομάδα στην οποία αγωνίζεται
CLUB	Σύλλογος στον οποίο αγωνίζεται
LEAGUE	Πρωτάθλημα στο οποίο συμμετέχει ο σύλλογος
POS	Θέση Ποδοσφαιριστή {GK, DF, MF, FW}
ROLE	Ρόλος Ποδοσφαιριστή (Ακριβής Θέση Ποδοσφαιριστή)
AGE_EX	Ηλικία Ποδοσφαιριστή
AGE_GR	Ηλικιακή Ομάδα Ποδοσφαιριστή {15-20, 21-25, 26-30, 31-35, 36+}
MATCHES	Συμμετοχές σε αγώνες του Παγκοσμίου Κυπέλλου 2018
GOALS	Γκολ σε αγώνες του Παγκοσμίου Κυπέλλου 2018
ASSISTS	Τελικές πάσες σε αγώνες του Παγκοσμίου Κυπέλλου 2018
Y-CARDS	Κίτρινες κάρτες σε αγώνες του Παγκοσμίου Κυπέλλου 2018
R-CARDS	Κόκκινες κάρτες σε αγώνες του Παγκοσμίου Κυπέλλου 2018
MINUTES	Λεπτά που αγωνίστηκε στο Παγκόσμιο Κύπελλο 2018
N-CAPS	Συνολικές συμμετοχές με την εθνική ομάδα
N-GOALS	Συνολικά Γκολ με την εθνική ομάδα

Πίνακας 4.1: Στήλες της βάσης δεδομένων

Το παραπάνω σετ δεδομένων καθίσταται ιδιαίτερα εύχρηστο καθώς περιλαμβάνει γνωρίσματα τα οποία συμβάλουν στην κατηγοριοποίηση των ποδοσφαιριστών που συμμετείχαν στην διοργάνωση με βάση διαφορετικά χαρακτηριστικά όπως η θέση στην οποία αγωνίζονται, η ηλικία τους και δεδομένα που αφορούν τις ομάδες που αγωνίζονται σε συλλογικό επίπεδο. Επί προσθέτως υπάρχουν γνωρίσματα που αντικατοπτρίζουν διαφορετικές πτυχές της παρουσίας τους στην διοργάνωση όπως η συμμετοχή τους (Αγώνες, Λεπτά), η συμβολή τους στο επιθετικό παιχνίδι της εθνικής τους ομάδας (Γκολ, Τελικές Πάσες) και η πειθαρχική τους συμπεριφορά (Κάρτες).

Τέλος θα πρέπει να σημειωθεί ότι για την εξαγωγή καλύτερων συμπερασμάτων από την αναλυτική επεξεργασία των δεδομένων θα πρέπει να χρησιμοποιηθεί μια βάση δεδομένων αρκετά μεγαλύτε-

ρου μεγέθους. Όμως η έλλειψη έτοιμων και ελεύθερων βάσεων δεδομένων στο διαδίκτυο που πληρούν τις προϋποθέσεις και έχουν τα κατάλληλα χαρακτηριστικά καθώς και ο απαιτούμενος χρόνος για την δημιουργία από την αρχή μιας μεγαλύτερης βάσης δεδομένων για να χρησιμοποιηθεί στα πλαίσια της εργασίας αυτής οδήγησε στην επιλογή της βάσης που παρουσιάστηκε προηγουμένως. Μεγαλύτερης σημασίας κρίνεται η μοντελοποίηση των δεδομένων και η δημιουργία μιας δομής που θα μπορεί να χρησιμοποιηθεί για την αναλυτική επεξεργασία δεδομένων ποδοσφαιρικών αγώνων εύκολα προσαρμόσιμη στα διαθέσιμα, την εκάστοτε στιγμή, δεδομένα.

4.2.1 Δημιουργία SQLite Table από αρχείο CSV

Η παραπάνω βάση δεδομένων έχει δημιουργηθεί στο Excel και βρίσκεται σε μορφή csv, όπως έχει προαναφερθεί το Cubes μπορεί να διαχειριστεί βάσεις δεδομένων διάφορων τύπων όμως όχι στην μορφή αυτή. Για αυτό το λόγο θα χρησιμοποιηθεί μια συνάρτηση του Cubes που ονομάζεται create_table_from_csv, η συγκεκριμένη συνάρτηση δημιουργεί από ένα αρχείο csv έναν πίνακα με τις επιθυμητές στήλες του csv χρησιμοποιώντας το σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων που θα επιλέξει ο χρήστης. Στην εργασία αυτή έχει επιλεγεί το SQLite, συνεπώς μετά από την εκτέλεση του παρακάτω κώδικα θα δημιουργηθεί ένα αρχείο sqlite το οποίο θα περιέχει τα δεδομένα σε έναν μοναδικό πίνακα (denormalized table).

Όπως φαίνεται παρακάτω η συνάρτηση create_table_from_csv έχει πέντε παραμέτρους, η πρώτη είναι η database engine που θα χρησιμοποιηθεί στην περίπτωση αυτή το SQLite, η δεύτερη είναι το αρχείο csv το οποίο περιέχει τα δεδομένα, η τρίτη είναι το όνομα του πίνακα που θα δημιουργηθεί, η τέταρτη είναι οι στήλες που θα δημιουργηθούν εν αντιστοιχία με τις στήλες του csv, για την καθεμιά δηλώνεται το όνομα που θα έχει και ο τύπος των δεδομένων που θα περιέχει, τέλος η πέμπτη παράμετρος “create_id” εφόσον αρχικοποιηθεί σε ”True” τότε θα δημιουργηθεί μια ακόμα στήλη ”id” που θα χρησιμοποιηθεί ως πρωτεύον κλειδί για τον πίνακα.

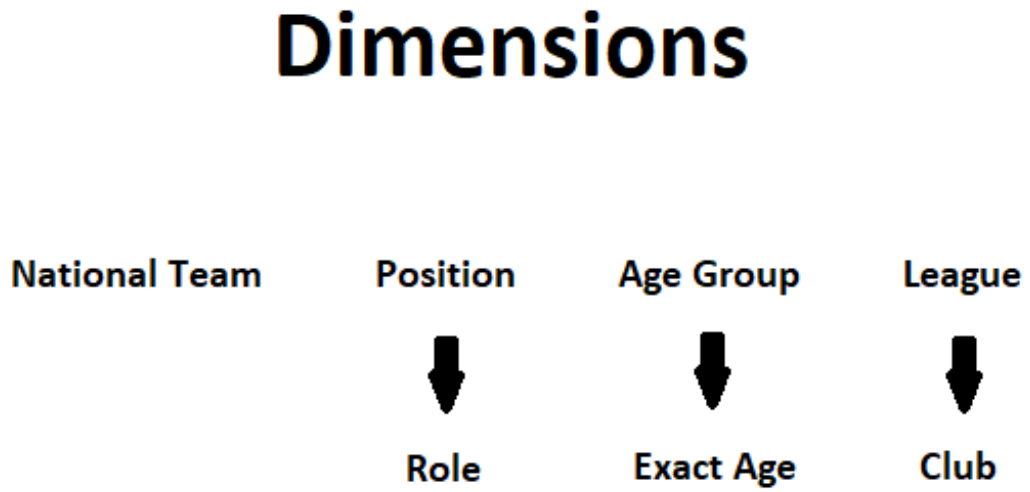
```
from sqlalchemy import create_engine
from cubes.tutorial.sql import create_table_from_csv

engine = create_engine('sqlite:///world_cup.sqlite')

create_table_from_csv(engine,
                      "WC18.csv",
                      table_name="players",
                      fields=[
                            ("name", "string"),
                            ("nat_team", "string"),
                            ("club", "string"),
                            ("league", "string"),
                            ("pos", "string"),
                            ("role", "string"),
                            ("age_gr", "string"),
                            ("age_ex", "string"),
                            ("matches", "integer"),
                            ("goals", "integer"),
                            ("assists", "integer"),
                            ("ycards", "integer"),
                            ("rcards", "integer"),
                            ("minutes", "integer"),
                            ("n-caps", "integer"),
                            ("n-goals", "integer")],
                      create_id=True
                  )
```

4.3 Δημιουργία λογικού μοντέλου

Για τα δεδομένα που παρουσιάστηκαν στην προηγούμενη υποενότητα έχει δημιουργηθεί ένα λογικό μοντέλο για να χρησιμοποιηθεί στην συνέχεια από το Cubes για την επεξεργασία τους με την προσέγγιση OLAP. Το μοντέλο αυτό θα περιέχει έναν μοναδικό (υπέρ-)κύβο τεσσάρων διαστάσεων, το κάθε στοιχείο του οποίου θα αντιπροσωπεύει έναν ποδοσφαιριστή, και θα περιέχει έξι μέτρα. Στο παρακάτω απόσπασμα κώδικα φαίνεται πως ορίζεται το μοντέλο σε αρχείο JSON, τα μεταδεδομένα που περιγράφονται είναι τα name, label, dimensions και cubes. Όλα τα υπόλοιπα μεταδεδομένα αρχικοποιούνται στις προκαθορισμένες τιμές τους.

```
{
  ”name”: ”world_cup_model”,
  ”label”: ”World Cup Russia 2018”,
  ”dimensions”: [...],
  ”cubes”: [...],
}
```

Ο υπερκύβος θα έχει τέσσερις διαστάσεις, η πρώτη μονοεπίπεδη διάσταση θα είναι η εθνική ομάδα στην οποία αγωνίζεται ο ποδοσφαιριστής (nat_team), η δεύτερη διάσταση θα είναι αυτή της ομάδας του ποδοσφαιριστή σε συλλογικό επίπεδο και θα έχει δύο επίπεδα, πρώτο το πρωτάθλημα στο οποίο αγωνίζεται (league) και δεύτερο τον σύλλογο στον οποίο αγωνίζεται (club). Η τρίτη διάσταση θα αφορά την θέση που αγωνίζεται ο ποδοσφαιριστής και θα είναι, όπως και η προηγούμενη, δύο επιπέδων με το πρώτο επίπεδο να αναφέρεται στην γενική θέση (pos) δηλαδή εάν αγωνίζεται στο τέρμα, στην άμυνα, στο κέντρο ή στην επίθεση, και το δεύτερο επίπεδο στην ακριβή θέση (role). Τέλος η τέταρτη διάσταση θα είναι και αυτή διεπίπεδη και αφορά την ηλικία του ποδοσφαιριστή με πρώτο επίπεδο την ηλιακή κατηγορία στην οποία βρίσκεται ο ποδοσφαιριστής (age_gr) και δεύτερο επίπεδο την ακριβή ηλικία του (age_ex). Περαιτέρω για τις τελευταίες δύο διαστάσεις ορίζονται δύο διαφορετικές ιεραρχίες τον επιπέδων τους, η πρώτη (default) θα έχει προφανώς τα δύο επίπεδα με την σειρά που αυτά έχουν καθοριστεί προηγουμένως ενώ η δεύτερη ιεραρχία για την διάσταση της θέσης θα παραλείπει την γενική θέση και θα περιέχει μόνο το επίπεδο της ακριβής θέσης, ενώ για την διάσταση της ηλικίας θα παραλείπει την ηλικιακή ομαδοποίηση και θα περιέχει μόνο το επίπεδο της ακριβής ηλικίας, συνεπώς θα «μετατρέπονται» οι διαστάσεις αυτές σε μονοεπίπεδες. Ακολουθεί ο ορισμός των μεταδεδομένων για τις διαστάσεις που προαναφέρθηκαν.



Σχήμα 4.1: Οι τέσσερις διαστάσεις του μοντέλου

```
”dimensions”: [
{
  ”name”:”nat_team”,
  ”label”:”National Team”
},{
  ”name”:”team”,
  ”levels”:[{
    ”name”:”league”,
    ”label”:”League”
  },{
    ”name”:”club”,
    ”label”:”Club”
  } ]
},{
  ”name”:”position”,
  ”levels”:[{
    ”name”:”pos”,
    ”label”:”Position”
  },{
    ”name”:”role”,
    ”label”:”Role”
  } ]
},{
  ”name”:”age”,
  ”levels”:[{
    ”name”:”age_gr”,
    ”label”:”Age Group”
  },{
    ”name”:”age_ex”,
    ”label”:”Exact Age”
  } ],
  ”hierarchies”:[{
    ”name”:”default”,
    ”levels”:[”age_gr”,”age_ex”]
  },{
    ”name”:”only_age”,
    ”levels”:[”age_ex”]
  }]
}]
```

Όπως αναφέρθηκε και προηγουμένως το μοντέλο θα περιέχει έναν μοναδικό κύβο ο οποίος θα έχει ως στοιχεία τους 736 ποδοσφαιριστές που συμμετείχαν στην τελική φάση του παγκοσμίου κυπέλλου του 2018. Ο κύβος θα ονομάζεται ”players” και θα περιέχει και τις τέσσερις διαστάσεις που ορίστηκαν προηγουμένως. Παρακάτω φαίνεται ο ορισμός του κύβου και τα μεταδεδομένα που αρχικοποιούνται. Οι ορισμοί των measures, aggregates, details και mappings θα παρουσιαστούν στην συνέχεια.

```
”cubes”: [
{
  ”name”: ”players”,
  ”label”: ”Players”,
  ”dimensions”: [”nat_team”,”team”,”position”,”age”],
  ”measures”: [......],
  ”aggregates”: [......],
  ”details”: [......],
  ”mappings”: {......}
}]
```

Ο κύβος θα περιέχει έξι μέτρα, αυτά θα είναι τα έξι που περιγράφουν την παρουσία του ποδοσφαιριστή στην διοργάνωση δηλαδή, οι αγώνες που συμμετείχε, τα λεπτά που αγωνίστηκε, τα τέρματα που πέτυχε, οι τελικές πάσες που μοίρασε και οι κάρτες, κίτρινες ή κόκκινες που δέχθηκε.

```
”measures”: [{
  ”name”: ”matches”,
  ”label”: ”Matches Played”
},{
  ”name”: ”goals”,
  ”label”: ”Goals Scored”
},{
  ”name”: ”assists”,
  ”label”: ”Assists Given”
},{
  ”name”: ”ycards”,
  ”label”: ”Yellow Cards Taken”
},{
  ”name”: ”rcards”,
  ”label”: ”Red Cards Taken”
},{
  ”name”: ”minutes”,
  ”label”: ”Minutes Played”
}]
```

Όσον αφορά τις συναθροίσεις (aggregates) που θα περιέχει το μοντέλο πρέπει να οριστεί το ποιο μέτρο και ποια συνάρτηση χρησιμοποιείται για τον υπολογισμό καθεμίας από αυτές ή ποια έκφραση σε περίπτωση που η συνάθροιση διέπεται από σύνθετο χαρακτήρα. Παρακάτω φαίνεται πως ορίζεται ένα δείγμα από τις συνολικές συναθροίσεις που θα συμπεριληφθούν στο μοντέλο, με παρόμοιο τρόπο μπορεί να οριστεί οποιαδήποτε συνάθροιση με βάση τα μέτρα που ορίστηκαν παραπάνω. Αξίζει να σημειωθεί πως η τελευταία συνάθροιση που ορίζεται δεν χρησιμοποιεί κανένα μέτρο καθώς πρόκειται για έναν απλό μετρητή των ποδοσφαιριστών που πληρούν τα κριτήρια που τίθενται κάθε φορά.

```
"aggregates": [{
  "name": "Total Goals",
  "function": "sum",
  "measure": "goals"
},{
  "name": "Total Assists",
  "function": "sum",
  "measure": "assists"
},{
  "name": "Total Caps",
  "function": "sum",
  "measure": "matches"
},{
  "name": "Average Goals",
  "expression": "round(avg(goals),3)"
},{
  "name": "Max Goals",
  "function": "max",
  "measure": "goals"
},{
  "name": "Scored at least one",
  "expression": "sum(if((goals>0),1,0))",
},{
  "name": "players",
  "functions": "count"
}]
```

Στα μέτρα του μοντέλου δεν έχουν συμπεριληφθεί οι στήλες της βάσης δεδομένων που αφορούσαν τις συνολικές συμμετοχές και γκολ κάθε ποδοσφαιριστή με την φανέλα της εθνικής του ομάδας πριν από το παγκόσμιο κύπελλο, δηλαδή οι στήλες N-CAPS και N-GOALS. Αυτό συμβαίνει γιατί στην παρούσα εργασία η αναλυτική επεξεργασία γίνεται στα δεδομένα που αφορούν την παρουσία

των ποδοσφαιριστών στην συγκεκριμένη διοργάνωση, συνεπώς οι δύο στήλες αυτές έχουν συμπληρωματικό χαρακτήρα και προσφέρουν επιπρόσθετες πληροφορίες για κάθε ποδοσφαιριστή. Για τον λόγο αυτό τα N-CAPS, N-GOALS καθώς και τα ονόματα των ποδοσφαιριστών ορίζονται ως λεπτομέρειες (details). Οι λεπτομέρειες δεν παίρνουν μέρος στα σύνολα που υπολογίζονται αλλά εμφανίζονται από την συνάρτηση ”facts”, η οποία επιστρέφει πληροφορίες για τα στοιχεία που βρίσκονται στο κελί του κύβου που ορίζουμε κατά την αναλυτική επεξεργασία.

```
”details”: [{
  ”name”: ”n-caps”,
  ”label”: ”Total National Caps”
},{
  ”name”: ”n-goals”,
  ”label”: ”Total National Goals”
},{
  ”name”: ”name”,
  ”label”: ”Name”
}]
```

Τέλος θα πρέπει να οριστούν οι αντιστοιχίσεις (mappings) μεταξύ του λογικού μοντέλου με τις κατάλληλες στήλες των πινάκων της βάσης δεδομένων. Όπως περιγράφηκε αναλυτικότερα στην υποενότητα του προηγούμενου κεφαλαίου που αφορούσε το εσωτερικό μέρος SQL, η προκαθορισμένη λειτουργία αναμένει μια βάση δεδομένων σε σχήμα αστέρα με τα ονόματα των διαστάσεων να αντιστοιχούν στα details tables του αστέρα. Συνεπώς στην συγκεκριμένη περίπτωση το εσωτερικό μέρος θα περίμενε έναν πίνακα γεγονότων με όνομα ”players” ο οποίος θα περιέχει ως στήλες όλα τα μέτρα και τις λεπτομέρειες του μοντέλου και σε διαφορετικούς πίνακες τις διαστάσεις. Στην πραγματικότητα όμως η βάση δεδομένων είναι σε έναν μόνο πίνακα (denormalized) και οι διαστάσεις βρίσκονται ως στήλες στον πίνακα γεγονότων. Έτσι θα πρέπει να οριστούν ρητά οι αντιστοιχίσεις μόνο για τις διαστάσεις και τα επίπεδα τους καθώς αυτά είναι τα μοναδικά που δεν «συμφωνούν» με την προκαθορισμένη σύμβαση. Παρακάτω φαίνεται πως δηλώνονται οι αντιστοιχίσεις αυτές μέσα στο μοντέλο.

```
”mappings”: {
  ”team.league”: ”league”,
  ”team.club”: ”club”,
  ”position.pos”: ”pos”,
  ”position.role”: ”role”,
  ”age.age_gr”: ”age_gr”,
  ”age.age_ex”: ”age_ex”
}
```

Με αυτόν τον τρόπο ολοκληρώνεται το λογικό μοντέλο το οποίο θα εφαρμοστεί στην βάση δεδομένων που παρουσιάστηκε προηγουμένως για να μπορέσουμε να την επεξεργαστούμε αναλυτικά ως κύβο OLAP.

4.4 Αρχικοποίηση και λειτουργία εξυπηρετητή Slicer

Εφόσον έχει οριστεί το λογικό μοντέλο το Cubes παρέχει την δυνατότητα για άμεση αναλυτική επεξεργασία μέσω του HTTP Server που διαθέτει ενσωματωμένο, ο εξυπηρετητής αυτός ουσιαστικά προσφέρει ένα επίπεδο αφαίρεσης μεταξύ του χρήστη και της υλοποίησης του framework σε Python. Αν και στην συγκεκριμένη εργασία θα δημιουργηθεί ένα περιβάλλον για την αναλυτική επεξεργασία των δεδομένων από τον χρήστη απ' ευθείας σε Python αξίζει να παρουσιαστεί ο εξυπηρετητής Slicer και ο τρόπος που μπορεί να χρησιμοποιηθεί.

Αρχικά όπως αναφέρθηκε και στο προηγούμενο κεφάλαιο είναι αναγκαίο να δημιουργηθεί ένα αρχείο .ini στο οποίο θα τίθενται οι απαραίτητες παράμετροι για την αρχικοποίηση του Slicer, στο αρχείο αυτό θα πρέπει να δηλωθεί το αρχείο των δεδομένων που θέλουμε να επεξεργαστούμε και το αρχείο του μοντέλου που θέλουμε να εφαρμόσουμε πάνω σε αυτά. Στην περίπτωση που μελετάται τα δεδομένα είναι στο αρχείο .sqlite που δημιουργήθηκε κατά την προετοιμασία των δεδομένων και το μοντέλο αυτό που ορίστηκε στην προηγούμενη υποενότητα. Παρακάτω φαίνεται το αρχείο .ini που απαιτείται για την αρχικοποίηση του HTTP Server.

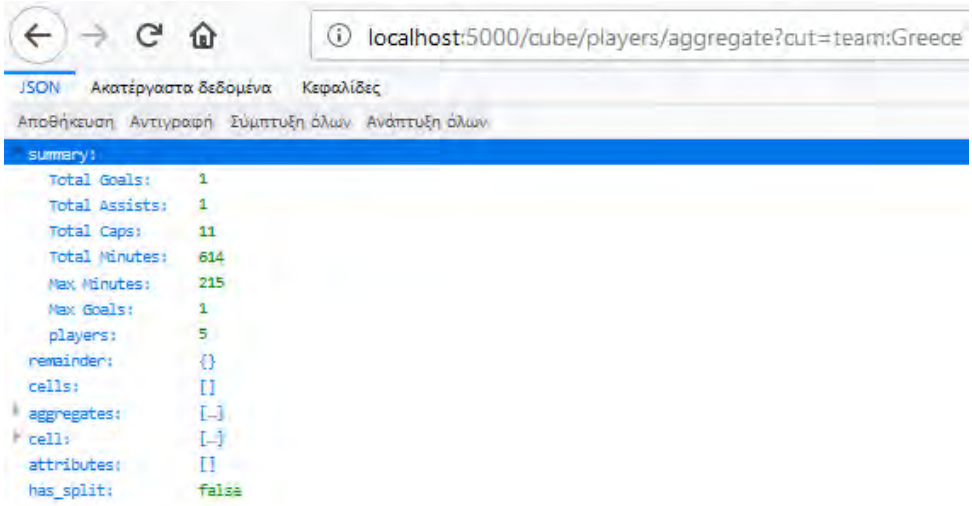
```
[store]
type: sql
url: sqlite:///data.sqlite

[models]
model: world_cup_model.json
```

Πλέον έχοντας ορίσει το αρχείο .ini αρκεί να δοθεί στην γραμμή εντολών η εντολή :

slicer serve slicer.ini

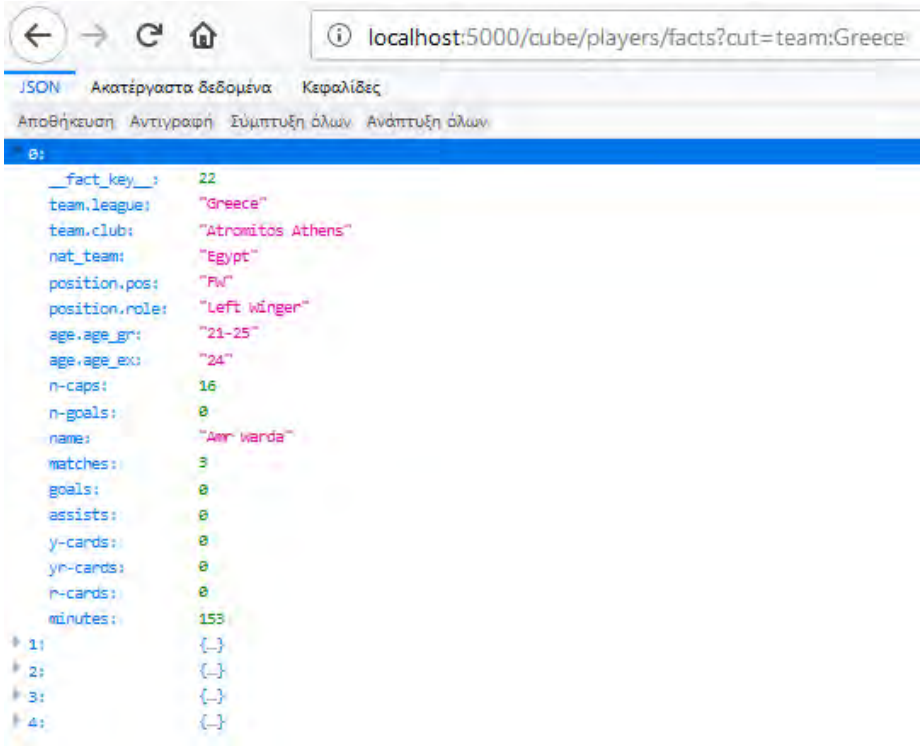
και στην συνέχεια ο χρήστης μπορεί να μεταβεί στην σελίδα **http://localhost:5000/cube/players/** όπου τρέχει ο Slicer και να υποβάλει τα ερωτήματα που θέλει στην μορφή HTTP Requests όπως αναλύθηκε στο προηγούμενο κεφάλαιο. Παρακάτω ακολουθούν ορισμένα παραδείγματα ερωτημάτων και τα αποτελέσματα που αυτά παράγουν.

Παράδειγμα πρώτο: Ζητούνται οι συναθροίσεις που έχουν οριστεί επιλέγοντας ως κελί το υποσύνολο των δεδομένων που αφορά ποδοσφαιριστές που αγωνίζονται στο ελληνικό πρωτάθλημα. Άρα απαιτείται μία τομή στην διάσταση "team.league", εφόσον το επίπεδο της "league" είναι το πρώτο επίπεδο της συγκεκριμένης διάστασης αρκεί να οριστεί ως εξής **cut=team:Greece**



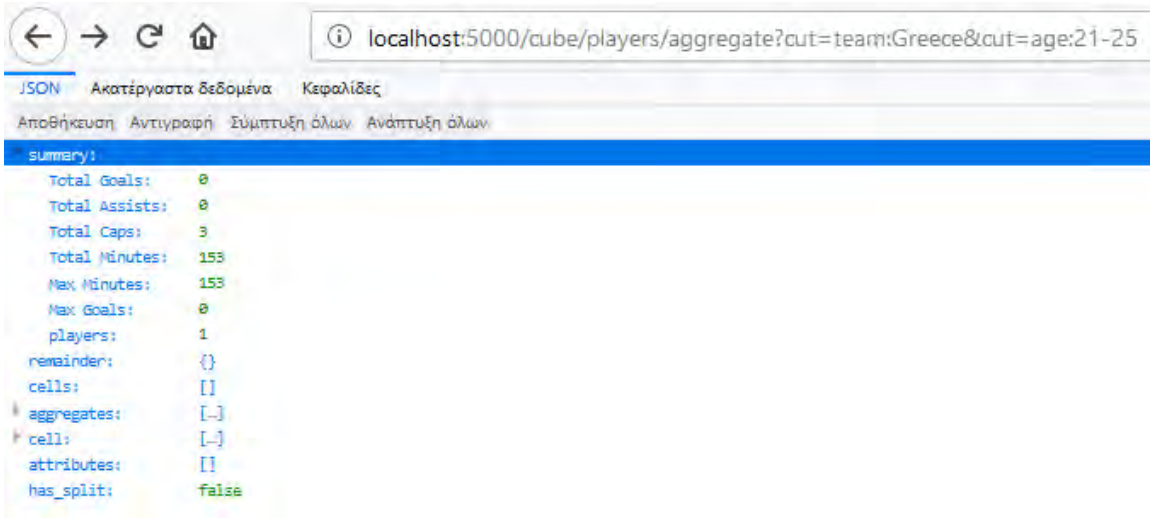
Όπως φαίνεται παραπάνω το αποτέλεσμα του ερωτήματος είναι σε μορφή JSON και περιέχει τις συναθροίσεις για τους πέντε ποδοσφαιριστές που πληρούν το κριτήριο.

Παράδειγμα δεύτερο: Ζητούνται τα στοιχεία (δηλαδή οι ποδοσφαιριστές) που ανήκουν στο κελί του προηγούμενου παραδείγματος.

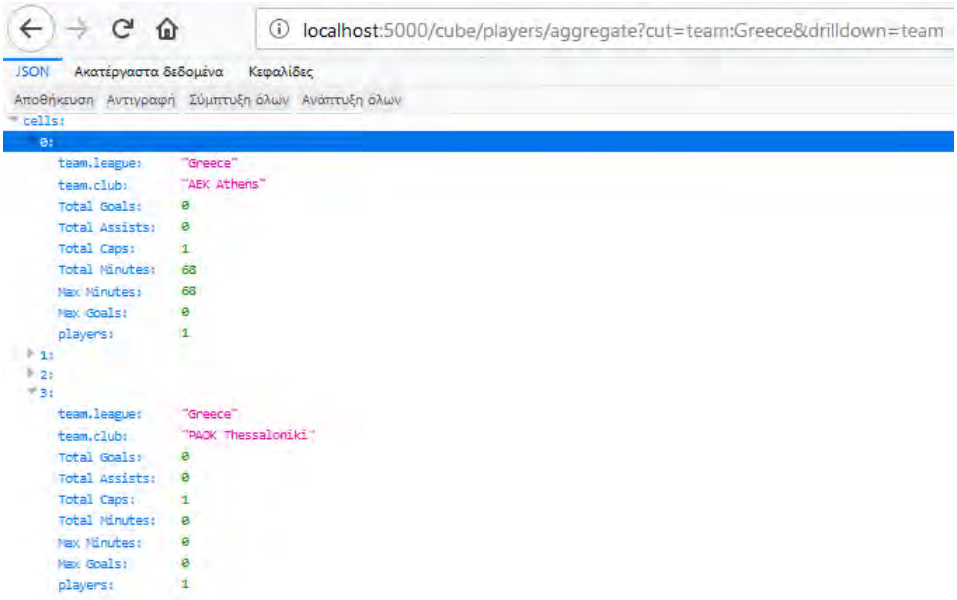


Το αποτέλεσμα είναι μια λίστα από τα πέντε στοιχεία που συνθέτουν το ζητούμενο κελί και όλα τα δεδομένα για το κάθε ένα από αυτά (για οικονομία χώρου στην παραπάνω εικόνα φαίνεται ανεπτυγμένο μόνο το πρώτο στοιχείο της λίστας).

Παράδειγμα τρίτο: Ζητούνται οι συναθροίσεις που έχουν οριστεί επιλέγοντας ως κελί το υποσύνολο των δεδομένων που αφορά ποδοσφαιριστές που αγωνίζονται στο ελληνικό πρωτάθλημα ΚΑΙ ανήκουν στην ηλικιακή ομάδα 21-25. Άρα πέρα από την τομή στην διάσταση ”team” που υπήρχε και στα δύο προηγούμενα παραδείγματα θα πρέπει να προστεθεί και μία τομή στην διάσταση ”age”.



Παράδειγμα τέταρτο: Ζητούνται οι συναθροίσεις που έχουν οριστεί επιλέγοντας ως κελί το υποσύνολο των δεδομένων που αφορά ποδοσφαιριστές που αγωνίζονται στο ελληνικό πρωτάθλημα ομαδοποιημένα κατά τον σύλλογο στον οποίο αγωνίζονται οι ποδοσφαιριστές. Συνεπώς θα πρέπει να προστεθεί και μία αναλυτική κάθοδος στην διάσταση ”team”.



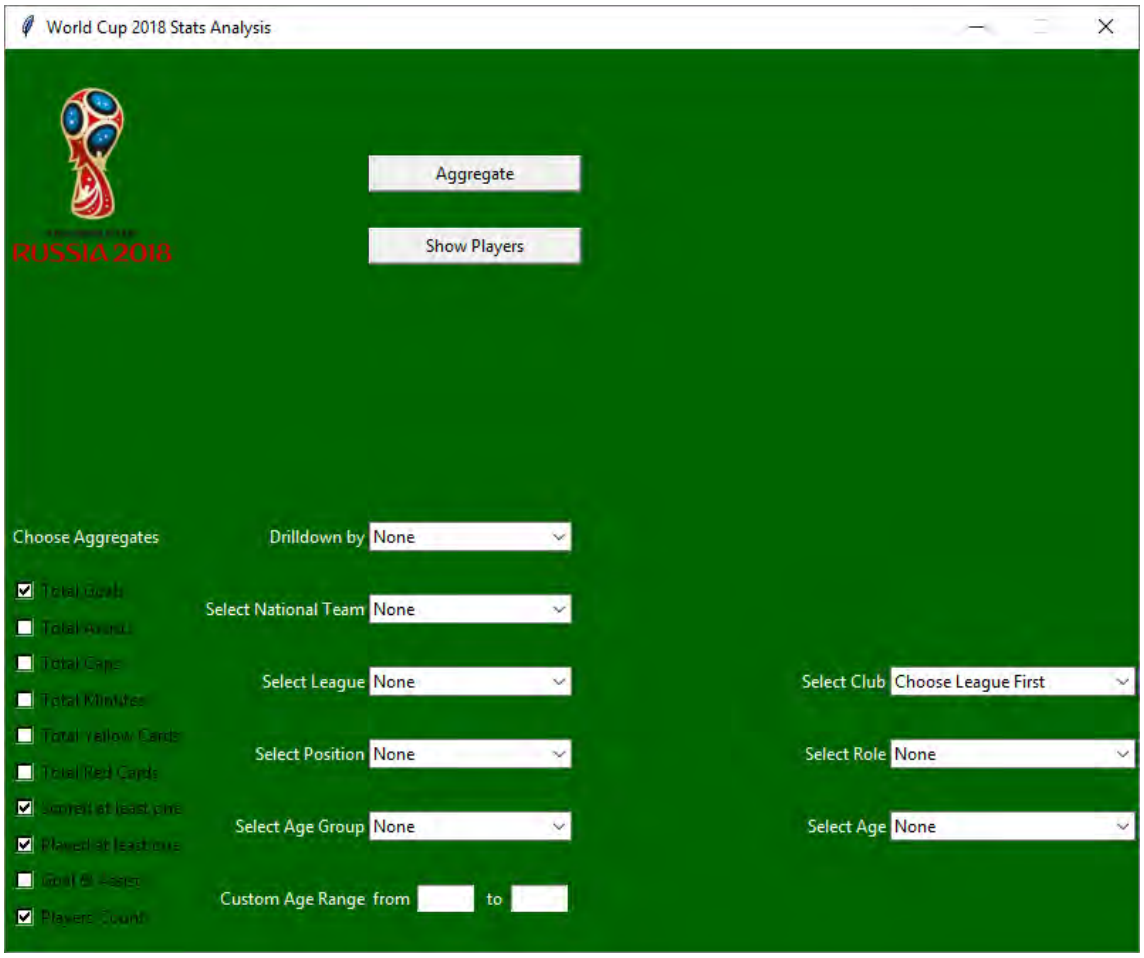
Στο αποτέλεσμα φαίνεται πως οι 5 ποδοσφαιριστές που αγωνίζονται στο ελληνικό πρωτάθλημα αγωνίζονται σε 4 διαφορετικές ομάδες.

4.5 Δημιουργία πλατφόρμας σε περιβάλλον Python

Στα πλαίσια της εργασίας αυτής έχει δημιουργηθεί μια πλατφόρμα για την άμεση αναλυτική επεξεργασία των δεδομένων του Παγκοσμίου Κυπέλλου 2018 στην προγραμματιστική γλώσσα Python χρησιμοποιώντας το framework Cubes για την επίτευξη της παραπάνω λειτουργικότητας. Παρακάτω θα παρουσιαστεί η εν λόγω πλατφόρμα τόσο από την προγραμματιστική οπτική της υλοποίησης της όσο και ο τρόπος λειτουργίας της και το πώς μπορεί να αξιοποιηθεί από τους χρήστες.

4.5.1 Παρουσίαση του GUI και τρόπος λειτουργίας

Στην παρακάτω εικόνα φαίνεται το γραφικό περιβάλλον χρήστη (GUI) της εφαρμογής

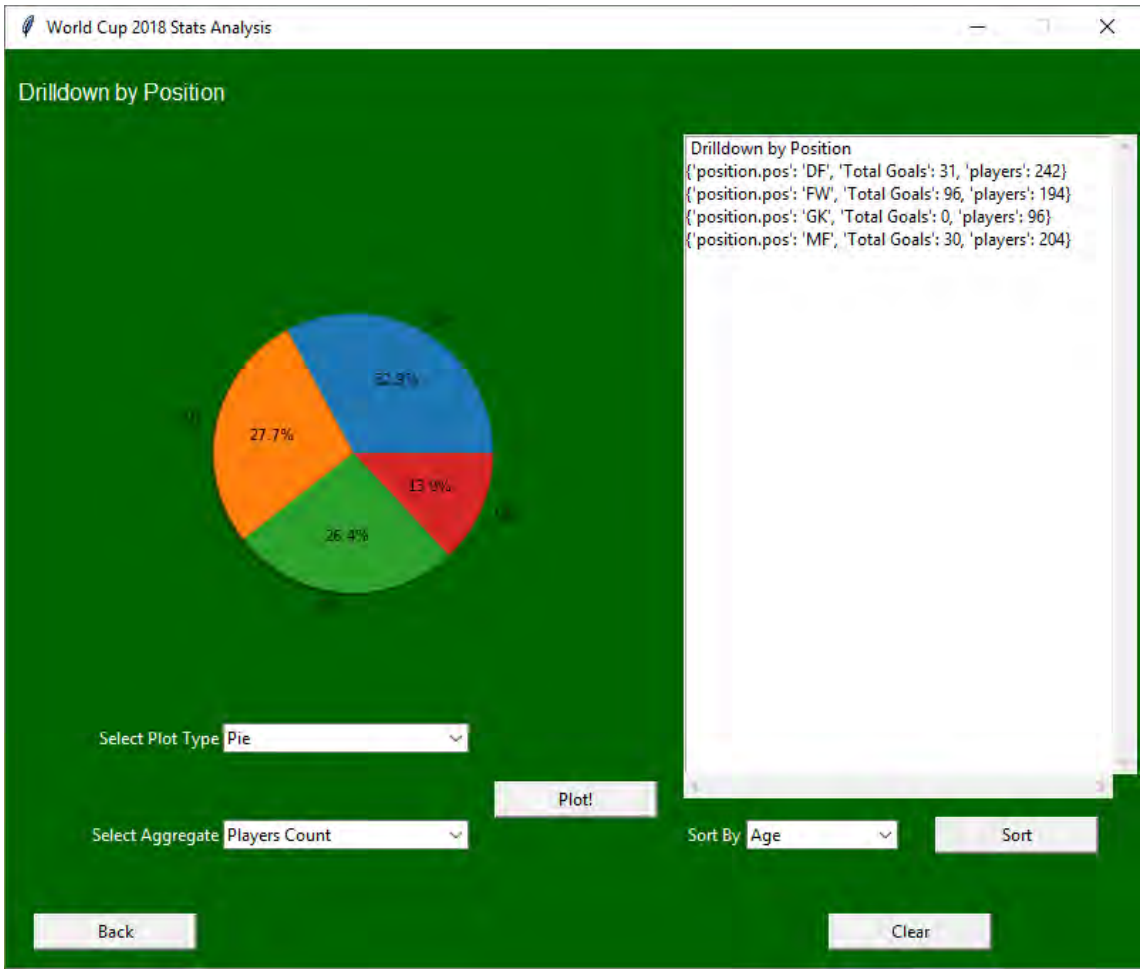


Σχήμα 4.2: Παράθυρο εισόδων του γραφικού περιβάλλοντος

Όπως φαίνεται παραπάνω στο επάνω τμήμα του γραφικού περιβάλλοντος υπάρχουν τα κουμπιά που εκτελούν τις δύο βασικές ενέργειες της εφαρμογής. Το κουμπί Aggregate οδηγεί στην παραγωγή των συνοπτικών στατιστικών και των συναθροίσεων που αντιστοιχούν στον υποκύβο που καθορίζεται από τον χρήστη, ενώ το κουμπί Show Players παράγει την λίστα με όλους τους ποδοσφαιριστές, δηλαδή τις εγγραφές που συνθέτουν τον υποκύβο αυτόν.

Το κάτω τμήμα του γραφικού περιβάλλοντος είναι αυτό στο οποίο ο χρήστης μπορεί να αρχικοποιήσει τις κατάλληλες παραμέτρους για τον τεμαχισμό και το κομμάτιασμα (slicing & dicing), δηλαδή τον καθορισμό του υποκύβου, καθώς και για την εκτέλεση αναλυτικής καθόδου (drill-down). Στο αριστερό τμήμα ο χρήστης μπορεί να τσεκάρει όλα τα στατιστικά και τις συναθροίσεις για τις οποίες ενδιαφέρεται και επιθυμεί να συμπεριληφθούν στο αποτέλεσμα. Στην συνέχεια με την επιλογή Drilldown by μπορεί να επιλέξει την διάσταση ως προς την οποία επιθυμεί να κατηγοριοποιήσει τα αποτελέσματα, δηλαδή να εκτελέσει αναλυτική κάθοδο. Επίσης υπάρχουν επτά Select τα οποία αφορούν το slicing & dicing και ορίζουν Point Cuts στις αντίστοιχες διαστάσεις ή στα αντίστοιχα επίπεδα διαστάσεων. Τέλος υπάρχει η επιλογή Custom Age Range με την οποία ο χρήστης μπορεί να ορίσει το ειδικό εύρος ηλικιών για το οποίο ενδιαφέρεται εκτελώντας Range Cut στην διάσταση της ηλικίας, αγνοώντας τελείως το επίπεδο Age Group και τα προκαθορισμένα εύρη ηλικιών που υπάρχουν διαθέσιμα.

Εφόσον ο χρήστης θέσει τις παραμέτρους για την ανάλυση που επιθυμεί να πραγματοποιήσει και πατήσει ένα από τα δύο κουμπιά που οδηγούν στην παραγωγή του αποτελέσματος θα οδηγηθεί στο παράθυρο που φαίνεται στην εικόνα που ακολουθεί.



Σχήμα 4.3: Παράθυρο αποτελεσμάτων του γραφικού περιβάλλοντος

Το παράθυρο των αποτελεσμάτων έχει δύο τμήματα, στα δεξιά εμφανίζονται τα αποτελέσματα των ερωτημάτων που έχει υποβάλει ο χρήστης. Στο αριστερό τμήμα, εάν έχει ζητηθεί αναλυτική κάθοδος κατά κάποια διάσταση τότε παρουσιάζεται ένα διάγραμμα πίτας το οποίο δείχνει το ποσοστό των ποδοσφαιριστών που εμπίπτουν στην εκάστοτε κατηγορία της διάστασης κατά την οποία εκτελείται η αναλυτική κάθοδος όπως φαίνεται και στην προηγούμενη εικόνα. Στην περίπτωση που δεν έχει ζητηθεί αναλυτική κάθοδος τότε στην θέση της πίτας θα εμφανιστεί ένας πίνακας στατιστικών ο οποίος περιλαμβάνει σύνολα, μέγιστους και μέσους όρους για όλα τα μέτρα του κύβου. Η μορφή του πίνακα στατιστικών αυτού φαίνεται στην παρακάτω εικόνα. Και στις δύο περιπτώσεις ο χρήστης μπορεί να ζητήσει κάποιο διαφορετικό διάγραμμα, όπως για παράδειγμα ραβδόγραμμα ή θηκόγραμμα, επιλέγοντας από την λίστα διαγραμμάτων, καθορίζοντας την συνάθροιση την οποία επιθυμεί να αποτυπωθεί στο διάγραμμα αυτό και πατώντας στο κουμπί Plot.

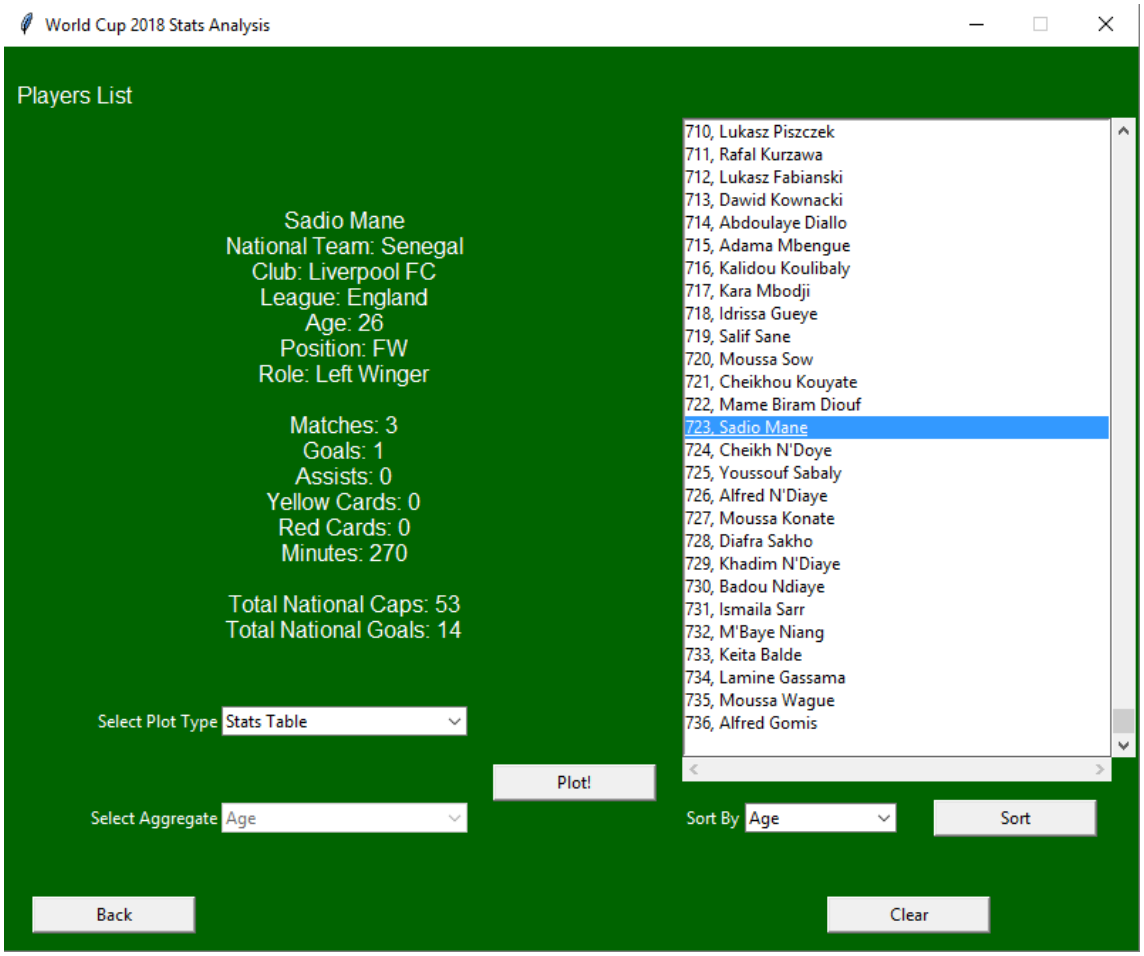
	Total	Max	Average
Goals	157.0	6.0	0.213
Assists	125.0	3.0	0.17
Caps	1790.0	7.0	2.432
Minutes	129846.0	694.0	176.421
Yellow Cards	219.0	3.0	0.298
Red Cards	4.0	1.0	0.005

Σχήμα 4.4: Πίνακας Στατιστικών

Τα αποτελέσματα του νέου ερωτήματος θα προστεθούν στο τέλος της στήλης αποτελεσμάτων χωρίς να σβηστούν τα αποτελέσματα των ερωτημάτων που έχουν προηγηθεί έτσι ώστε ο χρήστης να μπορεί να συγκρίνει τα αποτελέσματα μεταξύ τους, ενώ αν επιθυμεί τον καθαρισμό της λίστας αυτό επιτυγχάνεται με το κουμπί Clear. Επιπλέον εάν ο χρήστης πατήσει πάνω στο όνομα κάποιου ποδοσφαιριστή στην στήλη αποτελεσμάτων, έπειτα από την εκτέλεση της λειτουργίας Show Players, τότε θα εμφανιστούν στο αριστερό τμήμα της οθόνης περισσότερες πληροφορίες για τον συγκεκριμένο ποδοσφαιριστή και οι επιδόσεις του στην διοργάνωση. Η τελευταία λειτουργία φαίνεται στην παρακάτω εικόνα.

Επίσης κάτω από την στήλη αποτελεσμάτων υπάρχει το κουμπί Sort με το οποίο ο χρήστης μπορεί

να ζητήσει την ταξινόμηση των ποδοσφαιριστών που ανήκουν στο υποσύνολο του κύβου που έχει καθοριστεί από το slicing & dicing κατά το μέτρο που θα επιλέξει από την αντίστοιχη λίστα. Η εμφάνιση της κατάταξης θα εμφανιστεί στην στήλη αποτελεσμάτων ακριβώς από πάνω, δίπλα σε κάθε ποδοσφαιριστή θα εμφανίζεται και το ποσοστό των ποδοσφαιριστών που έχουν ίδια ή μεγαλύτερη επίδοση στο συγκεκριμένο χαρακτηριστικό. Τέλος κάτω αριστερά υπάρχει το κουμπί Back το οποίο οδηγεί στην προηγούμενη οθόνη για να υποβάλει ο χρήστης το επόμενο ερώτημα.



Σχήμα 4.5: Προβολή στοιχείων ποδοσφαιριστή

4.5.2 Παρουσίαση του κώδικα

Για την δημιουργία της παραπάνω εφαρμογής χρησιμοποιήθηκε η βιβλιοθήκη του Cubes που προσφέρει την λειτουργικότητα OLAP πάνω στα δεδομένα και η βιβλιοθήκη Tkinter για την δημιουργία του γραφικού περιβάλλοντος. Επίσης για την απεικόνιση των διαγραμμάτων χρησιμοποιήθηκε η συλλογή PyPlot της βιβλιοθήκης Matplotlib, καθώς και σε μικρότερο βαθμό η βιβλιοθήκη NumPy, για την διαχείριση των πινάκων και η βιβλιοθήκη Pandas για τον χειρισμό δεδομένων.

```
 7 # IMPORTS ##################################
 8 from tkinter import *
 9 from cubes import *
10 import matplotlib.pyplot as plt
11 import numpy as np
12 import pandas as pd
13 from tkinter import ttk
14 ##########################################
```

Το πρώτο απαραίτητο βήμα είναι η αρχικοποίηση του Cubes ορίζοντας τον χώρο εργασίας (Workspace), δηλώνοντας την βάση δεδομένων που θα χρησιμοποιηθεί ως πυρήνας της αναλυτικής επεξεργασίας και το λογικό μοντέλο που θα εφαρμοστεί πάνω σε αυτήν. Έπειτα αρχικοποιείται ο περιηγητής συναθροίσεων δηλώνοντας τον κύβο του λογικού μοντέλου τον οποίο θα χρησιμοποιήσει.

```
18 # CUBES INITIALIZATION#######################
19 ## WORKSPACE
20 ws = Workspace()
21 ws.register_default_store("sql", url="sqlite:///world_cup.sqlite")
22 ws.import_model("world_cup_model.json")
23
24 ## BROWSER
25 browser = ws.browser("players")
26 ################################################
```

Το Tk είναι ένα εργαλείο ανοικτού κώδικα που περιλαμβάνει τα απαραίτητα στοιχεία για την δημιουργία γραφικού περιβάλλοντος χρήστη σε πληθώρα προγραμματιστικών γλωσσών. Στην περίπτωση της Python η βιβλιοθήκη που απαιτείται για την χρήση του είναι η Tkinter (Tk Interface). Παρακάτω φαίνεται πως αρχικοποιείται το παράθυρο γραφικού περιβάλλοντος της εφαρμογής ενώ στην συνέχεια θα παρουσιαστούν όλα τα στοιχεία που υπάρχουν μέσα σε αυτό.

```
830 Graphical User Interface
831 '''
832 ###############################################################################
833 root = Tk()
834 root.configure(background='dark green')
835 root.title("World Cup 2018 Stats Analysis")
836 root.resizable(False, False)
```

...

```
1163
1164 root.mainloop()
1165 ###############################################################################
```

Όπως παρουσιάστηκε και στην προηγούμενη υποενότητα, το γραφικό περιβάλλον αποτελείται από δύο πλαίσια όσον αφορά την εμφάνιση του παραθύρου, το πρώτο πλαίσιο είναι αυτό στο οποίο ο

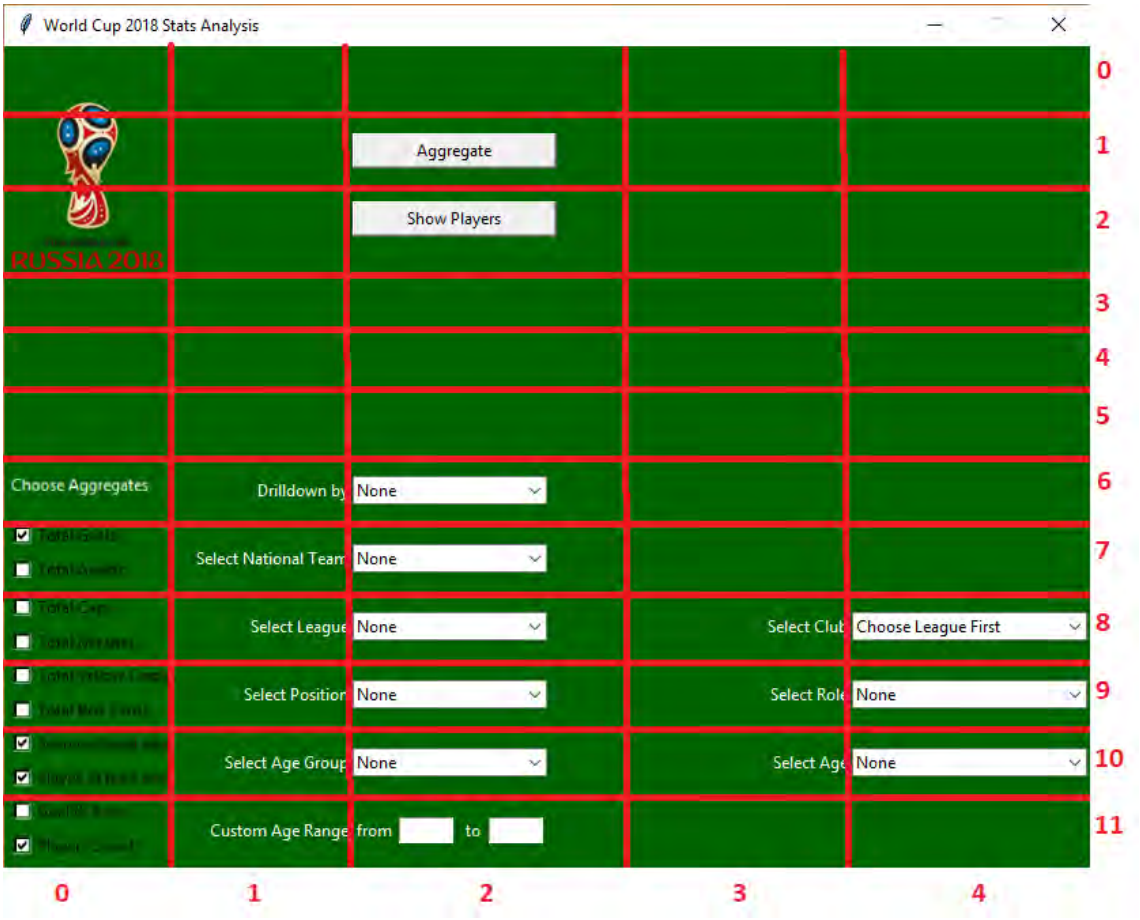
χρήστης δίνει τις επιλογές και ορίζει το ερώτημα που θέλει να υποβάλει ενώ το δεύτερο πλαίσιο είναι αυτό στο οποίο εμφανίζονται τα αποτελέσματα του ερωτήματος. Για το κάθε ένα από τα δύο αυτά πλαίσια (frames) δημιουργείται ένας νοητός κάνναβος (grid) ο οποίος θα έχει 12 γραμμές και 5 στήλες, μοναδικός ρόλος του κάνναβου αυτού είναι να βοηθήσει στην τοποθέτηση των διαφόρων αντικειμένων μέσα στο παράθυρο.

```
839 Frame1 = Label(root,bg='dark green')
840 Frame1.grid(row=0, column=0,rowspan=12,columnspan=5)
841
842 Frame2 = Frame(root,bg='dark green')
843 Frame2.grid(row=0, column=0,rowspan=12,columnspan=5)
844
845 raise_frame(Frame1)
846
847 # CONFIGURE 12x5 GRID FOR FRAME 1#######
848 rows = 0
849 while rows < 12:
850     Frame1.rowconfigure(rows, weight=0, minsize=51)
851     rows += 1
852 cols = 0
853 while cols < 5:
854     Frame1.columnconfigure(cols,weight=0, minsize=127)
855     cols += 1
856 ################################################
857 # CONFIGURE 12x5 GRID FOR FRAME 2######
858 rows = 0
859 while rows < 12:
860     Frame2.rowconfigure(rows, weight=0, minsize=50)
861     rows += 1
862 cols = 0
863 while cols < 5:
864     Frame2.columnconfigure(cols,weight=0, minsize=151)
865     cols += 1
866 ################################################
```

Η συνάρτηση raise_frame που χρησιμοποιείται στο προηγούμενο απόσπασμα είναι αυτή με την οποία γίνεται εναλλαγή μεταξύ των δύο πλαισίων όταν αυτό απαιτείται κατά την λειτουργία της εφαρμογής. Στην αρχή δίνεται η εντολή raise_frame(Frame1) έτσι ώστε η εφαρμογή να ξεκινάει με το πρώτο πλαίσιο. Παρακάτω φαίνεται πως ορίζεται η συγκεκριμένη συνάρτηση.

```
60 ################################################
61 def raise_frame(frame):
62     frame.tkraise()
63 ################################################
```

Πριν παρουσιαστεί ο κώδικας με τον οποίο ορίζονται τα στοιχεία που υπάρχουν στο πρώτο πλαίσιο (Frame1), στην παρακάτω εικόνα φαίνεται η τελική μορφή του πλαισίου όπως παρουσιάστηκε στην προηγούμενη υποενότητα με την προσθήκη του νοητού πλέγματος έτσι ώστε να είναι περισσότερο ξεκάθαρος ο τρόπος με τον οποίο τοποθετούνται τα στοιχεία μέσα σε αυτό.



Σχήμα 4.6: Κάνναβος παραθύρου εισόδων

Στο επάνω μέρος του πλαισίου υπάρχουν τρία στοιχεία, το πρώτο είναι το λογότυπο της διοργάνωσης του παγκοσμίου κυπέλλου του 2018, που αποτελεί την διοργάνωση που βρίσκεται στο επίκεντρο της ανάλυσης της εργασίας αυτής. Επίσης υπάρχουν δύο κουμπιά τα οποία εκτελούν τις βασικές λειτουργίες του περιηγητή του Cubes δηλαδή τα κουμπιά ”Aggregate” και ”Show Players”. Τα κουμπιά αυτά ενεργοποιούν τις συναρτήσεις aggregate και facts οι οποίες θα αναλυθούν αργότερα στο κείμενο. Ακολουθεί ο ορισμός των στοιχείων που προαναφέρθηκαν.

```
873 # LOGO ########################################
874 panel_logo = Label(Frame1)
875 panel_logo.configure(background='dark green')
876 panel_logo.grid(row=0,column=0,rowspan=3,columnspan=1,sticky=W)
877 logo = PhotoImage(file="logo_rus18.png")
878 panel_logo.config(image=logo)
879 ############################################
881 # BUTTONS ###################################
882 but1 = Button(Frame1, text="Aggregate",fg="black",height = 1, width = 20)
883 but1.bind("<Button-1>",aggregate)
884 but1.grid(row=1,column=2,sticky=W)
885
886 but2 = Button(Frame1, text="Show Players",fg="black",height = 1, width = 20)
887 but2.bind("<Button-1>",facts)
888 but2.grid(row=2,column=2,sticky=W)
889 ############################################
```

Στο κάτω και αριστερά μέρος του παραθύρου υπάρχει μια λίστα από checkboxes με τα οποία ο χρήστης μπορεί να επιλέξει ποιες από τις συναθροίσεις επιθυμεί να συμπεριληφθούν στο αποτέλεσμα. Σε κάθε ένα από αυτά αντιστοιχίζεται μια μεταβλητή η οποία παίρνει την τιμή 1 εάν το αντίστοιχο κουτί είναι επιλεγμένο και την τιμή 0 διαφορετικά. Στην αρχή οι μόνες συναθροίσεις που είναι επιλεγμένες είναι οι “Total Goals”, “Scored”, ”Played” και “Players Count”, για αυτό και οι αντίστοιχες μεταβλητές αρχικοποιούνται στο 1.

```
1003 # CHECKBOXES ################################
1004 var_goals = IntVar(value=1)
1005 Checkbutton(Frame1,text="Total Goals", variable=var_goals,
1006             background='dark green').grid(row=7,column=0, sticky=W+N)
1007
1008 var_assists = IntVar()
1009 Checkbutton(Frame1, text="Total Assists", variable=var_assists,
1010             background='dark green').grid(row=7,column=0, sticky=W+S)
1011
1012 var_caps = IntVar()
1013 Checkbutton(Frame1, text="Total Caps", variable=var_caps,
1014             background='dark green').grid(row=8,column=0, sticky=W+N)
1015
1016 var_minutes = IntVar()
1017 Checkbutton(Frame1, text="Total Minutes", variable=var_minutes,
1018             background='dark green').grid(row=8,column=0, sticky=W+S)
1019
1020 var_ycards = IntVar()
1021 Checkbutton(Frame1, text="Total Yellow Cards", variable=var_ycards,
1022             background='dark green').grid(row=9,column=0, sticky=W+N)
1023
1024 var_rcards = IntVar()
1025 Checkbutton(Frame1, text="Total Red Cards", variable=var_rcards,
1026             background='dark green').grid(row=9,column=0, sticky=W+S)
1027
1028 var_sc_count = IntVar(value=1)
1029 Checkbutton(Frame1, text="Scored at least one", variable=var_sc_count,
1030             background='dark green').grid(row=10,column=0, sticky=W+N)
1031
1032 var_pl_count = IntVar(value=1)
1033 Checkbutton(Frame1, text="Played at least one", variable=var_pl_count,
1034             background='dark green').grid(row=10,column=0, sticky=W+S)
1035
1036 var_ga_count = IntVar()
1037 Checkbutton(Frame1, text="Goal & Assist", variable=var_ga_count,
1038             background='dark green').grid(row=11,column=0, sticky=W+N)
1039
1040 var_count = IntVar(value=1)
1041 Checkbutton(Frame1, text="Players Count", variable=var_count,
1042             background='dark green').grid(row=11,column=0, sticky=W+S)
1043 ##################################################
```

Επίσης για τον καθορισμό του τεμαχισμού και του κομματιάσματος (slicing & dicing) του κύβου από τον χρήστη καθώς και για τον καθορισμό της διάστασης υπό την οποία επιθυμείτε αναλυτική κάθοδος (drilldown) έχουν οριστεί 8 drop-down lists ως read only comboboxes, μια για τον καθορισμό της αναλυτικής καθόδου, μια για την μόνοεπίπεδη διάσταση της εθνικής ομάδας και δύο για κάθε μια από τις τρεις διεπίπεδες διαστάσεις δηλαδή αυτές του συλλόγου, της θέσης και της

ηλικίας. Για κάθε μία από τις drop-down lists αυτές έχει αντιστοιχιστεί μια μεταβλητή η οποία λαμβάνει την τιμή την οποία επιλέγει ο χρήστης από τις διαθέσιμες που βρίσκονται στην λίστα, στην αρχική κατάσταση όλες οι λίστες έχουν ως προεπιλογή το “None”.

Επί προσθέτως για την κάθε μια από τις τρεις μεταβλητές που αντιστοιχούν στο πρώτο επίπεδο των διεπίπεδων διαστάσεων έχει ανατεθεί μια callback συνάρτηση η οποία καλείται κάθε φορά που ο χρήστης αλλάζει την επιλογή από την λίστα και ανάλογα με την νέα επιλογή ρυθμίζονται οι διαθέσιμης επιλογές της λίστας για το δεύτερο επίπεδο. Για παράδειγμα εάν ο χρήστης επιλέξει για το επίπεδο της ηλικιακής ομάδας «26-30» τότε η drop-down list που αφορά το επίπεδο της ακριβής ηλικίας θα περιέχει μόνο τις επιλογές που βρίσκονται σε αυτό το διάστημα. Παρακάτω φαίνεται πως έχουν οριστεί οι drop-down lists για το position και το age group, με παρόμοιο τρόπο έχουν οριστεί και οι υπόλοιπες.

```
942 # POSITION SELECTION BOX ######################
943 OPTIONS_POS = ['None','DF','FW','GK','MF']
944
945 pos_var = StringVar(Frame1)
946 pos_var.set(OPTIONS_POS[0]) ##default sto None
947 pos_var.trace("w",position_callback)
948
949 w4 = ttk.Combobox(Frame1, textvariable=pos_var,
950                   values=OPTIONS_POS, state='readonly',width=20)
951 w4.grid(row=9,column=2,sticky=W)
952 ################################################
953
954 # AGE GROUP SELECTION BOX ######################
955 OPTIONS_AGROUP = ['None','15-20','21-25','26-30','31-35','36+']
956
957 agroup_var = StringVar(Frame1)
958 agroup_var.set(OPTIONS_AGROUP[0]) ##default sto None
959 agroup_var.trace("w",age_callback)
960
961 w5 = ttk.Combobox(Frame1, textvariable=agroup_var,
962                   values=OPTIONS_AGROUP, state='readonly',width=20)
963 w5.grid(row=10,column=2,sticky=W)
964 ################################################
```

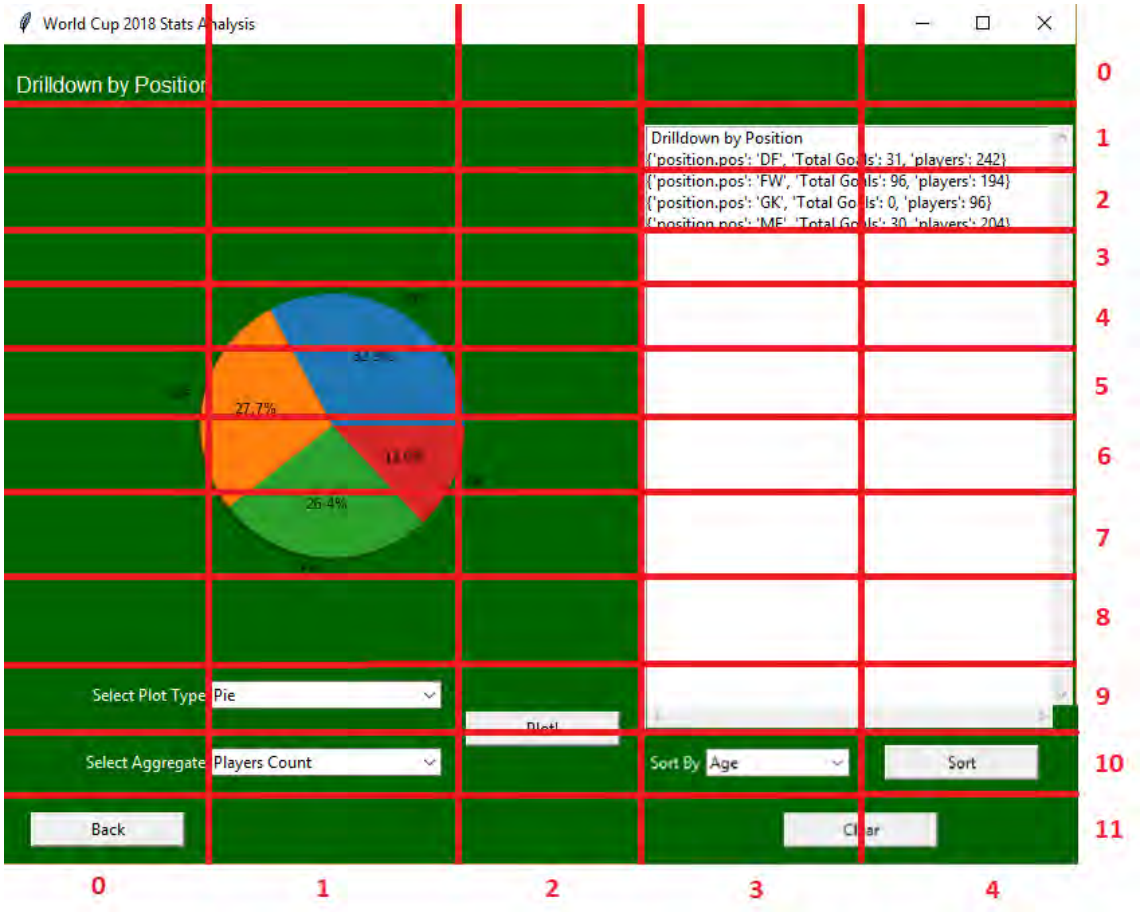
Στο παρακάτω απόσπασμα φαίνεται η συνάρτηση position_callback η οποία καλείται κάθε φόρα που ο χρήστης αλλάζει την μεταβλητή που έχει αντιστοιχιστεί στο combobox που αφορά την θέση και ανάλογα με την αλλαγή αυτή ορίζει τις διαθέσιμες επιλογές για το combobox που αφορά τον ακριβή ρόλο.

```
587 '''
588 Configure options for Role Combobox
589 '''
590 ##############################################################################
591 def position_callback(*args):
592 
593     global OPTIONS_ROLE
594     p = pos_var.get()
595     role_var.set('None')
596 
597     if p=='GK':
598         OPTIONS_ROLE = ['None','Goalkeeper']
599     elif p=='DF':
600         OPTIONS_ROLE = ['None','Centre-Back','Left-Back','Right-Back']
601     elif p=='MF':
602         OPTIONS_ROLE = ['None','Attacking Midfield','Central Midfield',
603                         'Defensive Midfield','Left Midfield','Right Midfield']
604     elif p=='FW':
605         OPTIONS_ROLE = ['None','Centre-Forward','Left Winger',
606                         'Right Winger','Second Striker']
607     else:
608         OPTIONS_ROLE = ['None','Centre-Back','Left-Back','Right-Back',
609                         'Centre-Forward','Left Winger','Right Winger',
610                         'Second Striker','Goalkeeper','Attacking Midfield',
611                         'Central Midfield','Defensive Midfield',
612                         'Left Midfield','Right Midfield']
613 
614     w6.config(values=OPTIONS_ROLE)
615 
616     return
617 ##############################################################################
```

Το τελευταίο στοιχείο που βρίσκεται στο πρώτο πλαίσιο είναι αυτό με το οποίο ο χρήστης μπορεί να εισάγει από το πληκτρολόγιο το ακριβές εύρος ηλικιών (custom age range) για το οποίο ενδιαφέρεται. Αυτό επιτυγχάνεται με την χρήση δύο εισόδων (Entries) οι οποίες αντιστοιχούν στα όρια του εύρους. Οι μεταβλητές που αντιστοιχίζονται σε αυτές τις εισόδους παίρνουν την τιμή που πληκτρολογεί ο χρήστης. Εάν ο χρήστης ορίσει μόνο το ένα από τα δύο όρια τότε το εύρος είναι ανοικτό και περιλαμβάνει όλες τις ηλικίες προς την κατεύθυνση που δεν έχει οριστεί όριο. Ο ορισμός αυτών των εισόδων φαίνεται παρακάτω.

```
1045 # ENTRIES ##########################################
1046 Lent = Label(Frame1,text="Custom Age Range",bg='dark green',fg='white')
1047 Lent.grid(row=11,column=1,sticky=E)
1048 
1049 Lent1 = Label(Frame1,text="from",bg='dark green',fg='white')
1050 Lent1.grid(row=11,column=2,sticky=W)
1051 Ent1 = Entry(Frame1,width=6)
1052 Ent1.grid(row=11,column=2,sticky=W,padx=34)
1053 Lent2 = Label(Frame1,text="to",bg='dark green',fg='white')
1054 Lent2.grid(row=11,column=2,padx=80,sticky=W)
1055 Ent2 = Entry(Frame1,width=6)
1056 Ent2.grid(row=11,column=2,padx=100,sticky=W)
1057 ##############################################
```

Όπως και προηγουμένως για το πρώτο πλαίσιο, παρακάτω παρουσιάζεται η τελική εικόνα του δεύτερου πλαισίου με τον νοητό κάνναβο που του αντιστοιχεί και στην συνέχει θα παρουσιαστεί ο τρόπος με τον οποίο ορίζονται τα στοιχεία που εμφανίζονται σε αυτόν.



Σχήμα 4.7: Κάνναβος παραθύρου αποτελεσμάτων

Καταρχάς κάτω αριστερά υπάρχει το κουμπί Back το οποίο οδηγεί στο προηγούμενο πλαίσιο χρησιμοποιώντας την συνάρτηση raise_frame που αναλύθηκε και προηγουμένως. Με την χρήση αυτού του κουμπιού ο χρήστης μπορεί να επιστρέψει στο προηγούμενο πλαίσιο για να υποβάλλει κάποιο νέο ερώτημα.

```
1125 but_back = Button(Frame2, width=15, text='Back',
1126                   command=lambda:raise_frame(Frame1))
1127 but_back.grid(row=11, column=0)
```

Όπως φαίνεται και παραπάνω υπάρχουν δύο drop-down lists με τις οποίες ο χρήστης καθορίζει τις παραμέτρους σύμφωνα με τις οποίες επιθυμεί να δημιουργήσει ένα διάγραμμα πατώντας το κουμπί Plot το οποίο ενεργοποιεί την αντίστοιχη συνάρτηση η οποία θα αναλυθεί αργότερα στο κείμενο.

Ο χώρος πάνω από τα στοιχεία αυτά είναι δεσμευμένος για την εμφάνιση των διαγραμμάτων ή για την εμφάνιση των πληροφοριών για κάποιον ποδοσφαιριστή. Καθώς ο ορισμός των στοιχείων αυτών είναι ανάλογος με τα αντίστοιχα στοιχεία ίδιου τύπου που υπήρχαν και στο προηγούμενο πλαίσιο δεν κρίνεται απαραίτητο να παρατεθούν τα αντίστοιχα αποσπάσματα κώδικα.

Στο δεξί μέρος του δεύτερου πλαισίου (Frame2) υπάρχει ο χώρος στον οποίο εκτυπώνονται τα αποτελέσματα των ερωτημάτων που υποβάλλει ο χρήστης. Επίσης εάν αυτά αφορούν αποτελέσματα της συνάρτησης ”facts” που αντιστοιχεί στο κουμπί Show Players ο χρήστης μπορεί να πατήσει πάνω στα αποτελέσματα αυτά και να δει περισσότερες πληροφορίες για τον αντίστοιχο ποδοσφαιριστή, για τον λόγο αυτό χρησιμοποιείται ένα listbox και όχι ένας απλός πίνακας για εκτύπωση χαρακτήρων. Επίσης στο listbox αυτό έχουν ανατεθεί δύο scrollbars ένα οριζόντιο και ένα κάθετο καθώς τα αποτελέσματα μπορεί να έχουν τόσο μεγάλο μήκος όσο και μεγάλο πλάτος. Τέλος η συνάρτηση listbox_select εκτελείτε κάθε φορά που ο χρήστης πατάει κλικ πάνω σε κάποιο αντικείμενο που υπάρχει στα αποτελέσματα. Ο ορισμός των παραπάνω στοιχείων φαίνεται στο απόσπασμα που ακολουθεί.

```
1147 # LISTBOX ######################################
1148 scrollver = Scrollbar(Frame2)
1149 scrollver.grid(row=1,column=5,rowspan=9,sticky=W+N+S)
1150 scrollhor = Scrollbar(Frame2,orient=HORIZONTAL)
1151 scrollhor.grid(row=10,column=3,columnspan=2,sticky=E+W+N)
1152
1153 listbox = Listbox(Frame2, yscrollcommand=scrollver.set,
1154                   xscrollcommand=scrollhor.set)
1155 listbox.grid(row=1,column=3,rowspan=9,columnspan=2,sticky=E+W+S+N)
1156
1157 scrollver.config(command=listbox.yview)
1158 scrollhor.config(command=listbox.xview)
1159
1160 listbox.bind('<<ListboxSelect>>', lambda e: select_listbox(e))
1161 ################################################
```

Το κουμπί Clear που βρίσκεται κάτω από την listbox εκτελεί την συνάρτηση ”clear” η οποία αφαιρεί όλα τα υπάρχοντα αντικείμενα μέσα από αυτήν. Ουσιαστικά σβήνει όλα τα αποτελέσματα των ερωτημάτων που έχουν παραχθεί μέχρι εκείνη την στιγμή. Τέλος τα κουμπιά Plot και Sort καλούν τις αντίστοιχες συναρτήσεις που θα αναλυθούν αργότερα.

```
66
67 ################################################
68 def clear():
69     listbox.delete('0','end')
70 ################################################
71
```

Εφόσον λοιπόν έχουν παρουσιαστεί όλα τα αντικείμενα που εμφανίζονται στο γραφικό περιβάλλον χρήστη, στην συνέχεια θα ακολουθήσουν τα αποσπάσματα του κώδικα που αφορούν τις βασικό-

τερες συναρτήσεις με την χρήση των οποίων επιτυγχάνεται η λειτουργικότητα του προγράμματος. Οι συναρτήσεις αυτές έχουν σαν πυρήνα την βιβλιοθήκη του Cubes.

Η συνάρτηση aggregate καλείτε όταν ο χρήστης πατάει το αντίστοιχο κουμπί και παράγει τα αποτελέσματα των συναθροίσεων που έχουν οριστεί. Αρχικά χρησιμοποιώντας τις συναρτήσεις drilldown_define, cell_define και aggr_define αρχικοποιούνται οι αντίστοιχοι παράμετροι και δίνονται ως είσοδοι στην συνάρτηση aggregate του browser του cubes. Στην συνέχεια τα αποτελέσματα που παράγονται από τον browser προστίθενται στην listbox του παραθύρου των αποτελεσμάτων, επίσης καλείτε η συνάρτηση plot για την δημιουργία του αντίστοιχου διαγράμματος. Τέλος καλείτε η συνάρτηση raise_frame για την μετάβαση του γραφικού περιβάλλοντος στο Frame2 όπου και εμφανίζονται τα αποτελέσματα. Ακολουθεί το απόσπασμα του αντίστοιχου κώδικα.

```
278 '''
279 Produce the result of Aggregation
280 '''
281 ###############################################################################
282 def aggregate(event):
283 
284     dr = drill_var.get()
285     a = drilldown_define()
286     b = cell_define()
287     c = aggr_define()
288 
289     result = browser.aggregate(cell=b,drilldown=a,aggregates=c)
290 
291     if a==None:
292         print(result.summary)
293         listbox.insert(END, result.summary)
294         plt_var.set(OPTIONS_PLOT[1])
295         plot()
296         label_title.config(text="\n Results ",font=40,fg='white')
297     else:
298         print("\n Drilldown by " + dr +" \n")
299         listbox.insert(END, "\n Drilldown by " + dr +" \n")
300 
301         for record in result:
302             print(record)
303             listbox.insert(END, record)
304 
305         plot()
306         label_title.config(text="\n Drilldown by " + dr +" \n",
307                            font=40,fg='white')
308 
309     raise_frame(Frame2)
310 
311 ###############################################################################
```

Οι συναρτήσεις drilldown_define και aggr_define είναι αρκετά απλές, η πρώτη αναθέτει στην παράμετρο που θα δοθεί ως είσοδο στον browser για την αναλυτική κάθοδο την τιμή που έχει επιλεχθεί από τον χρήστη στην αντίστοιχη drop-down list ενώ η δεύτερη δημιουργεί μια λίστα με όλες τις συναθροίσεις που έχουν επιλεχθεί στα αντίστοιχα checkboxes.

Η συνάρτηση cell_define δημιουργεί μια λίστα με όλες τις τομές στις διαστάσεις που έχουν επιλεχθεί από τον χρήστη από τις αντίστοιχες drop-down lists ή από τις εισόδους κειμένου εάν πρόκειται για το ηλικιακό εύρος. Στην συνέχεια δίνει ως είσοδο αυτή την λίστα στην συνάρτηση Cell του browser η οποία επιστρέφει τον υποκύβο που ορίζεται από τις τομές αυτές. Η συνάρτηση cell_define παρατίθεται στο ακόλουθο απόσπασμα.

```
463 '''
464 Define the cell of the cube
465 '''
466 ###############################################################################
467 def cell_define():
468 
469     cut=[]
470 
471     nateam = nateam_var.get()
472     if nateam!='None':
473         a=PointCut("nat_team",[nateam])
474         cut.append(a)
475 
476     league = league_var.get()
477     if league!='None':
478         b=PointCut("team",[league])
479         cut.append(b)
480 
            •••
501     role = role_var.get()
502     if role!='None':
503         f=PointCut("position",[role],hierarchy="only_role")
504         cut.append(f)
505 
506     custom_age1 = Ent1.get()
507     custom_age2 = Ent2.get()
508     if custom_age1!='' or custom_age2!='':
509         if len(custom_age1)==1: custom_age1= '0'+custom_age1
510         if len(custom_age2)==1: custom_age2= '0'+custom_age2
511         if custom_age1=='':custom_age1='15'
512         if custom_age2=='':custom_age2='45'
513         f=RangeCut("age",[custom_age1],[custom_age2],hierarchy="only_age")
514         cut.append(f)
515 
516     cell = Cell(browser.cube,cut)
517     return cell
518 ###############################################################################
```

Η συνάρτηση facts εκτελείται όταν ο χρήστης πατήσει το κουμπί “Show Players” και προσθέτει στην Listbox την λίστα με όλους τους ποδοσφαιριστές που είναι μέρη του υπο-κύβου που έχει καθοριστεί από τον χρήστη. Όπως και η συνάρτηση aggregate χρησιμοποιεί την συνάρτηση cell_define για τον καθορισμό του υπο-κύβου και στην συνέχεια καλεί την συνάρτηση facts του browser με είσοδο τον συγκεκριμένο υπο-κύβο. Στην συνέχεια προσθέτει στο τέλος της listbox την λίστα με τα ονόματα των ποδοσφαιριστών και το κλειδί που αντιστοιχεί στον καθένα. Τέλος καλεί την συνάρτηση plot() για να παραχθεί ο πίνακας των στατιστικών για τους ποδοσφαιριστές αυτούς.

```
314 '''
315 Print the facts included in the cell
316 '''
317 ################################################################################
318 def facts(event):
319 
320     a = cell_define()
321     result = browser.facts(cell=a)
322 
323     print("\n Players List: \n")
324     listbox.insert(END,"\n")
325     listbox.insert(END,"\n Players List: \n")
326 
327     rec_count=0
328     for record in result:
329         print(record["__fact_key__"],record["name"])
330         rec=str(record["__fact_key__"])+", "+record["name"]
331         listbox.insert(END, rec)
332         rec_count +=1
333 
334     if rec_count==0:
335         print("No player meets the criteria")
336         listbox.insert(END, "No player meets the criteria")
337 
338     drill_var.set(OPTIONS_DRILL[0])
339     plt_var.set(OPTIONS_PLOT[1])
340     plot()
341     label_title.config(text="\n Players List ",font=40,fg='white')
342     raise_frame(Frame2)
343 
344 ################################################################################
```

Η συνάρτηση select_Listbox καλείται κάθε φορά που ο χρήστης πατάει πάνω στο όνομα κάποιου ποδοσφαιριστή που έχει προστεθεί στην Listbox από την προηγούμενη συνάρτηση. Σκοπός της είναι να εμφανίσει στον χώρο των διαγραμμάτων τα πλήρη στοιχεία και στατιστικά του συγκεκριμένου ποδοσφαιριστή, διαβάζει από την Listbox το κλειδί που αντιστοιχεί στον ποδοσφαιριστή και στην συνέχεια τον εντοπίζει στο αποτέλεσμα της συνάρτησης facts του browser και εμφανίζει στην οθόνη όλα τα πεδία που τον αφορούν. Η συνάρτηση select_listbox παρατίθεται στο ακόλουθο απόσπασμα.

```
'''
Print details for player selected
'''
################################################################################
def select_listbox(event):
    w = event.widget
    try:
        idx = int(w.curselection()[0])
    except IndexError:
        return

    if w.get(idx)[0]!='{' and w.get(idx)[0]!='\n':
        key=w.get(idx).split(',')[0]

        result = browser.facts()

        for record in result:
            if record["__fact_key__"]==int(key) :

                fact=(record["name"]+"\n"+"National Team: "+
                      record["nat_team"]+"\n"+
                   "Club: "+record["team.club"]+"\n"+
                   "League: "+record["team.league"]+"\n"+
                   "Age: "+record["age.age_ex"]+"\n"+
                   "Position: "+record["position.pos"]+"\n"+
                   "Role: "+record["position.role"]+"\n"+"\n"+
                   "Matches: "+str(record["matches"])+"\n"+
                   "Goals: "+str(record["goals"])+"\n"+
                   "Assists: "+str(record["assists"])+"\n"+
                   "Yellow Cards: "+str(record["ycards"])+"\n"+
                   "Red Cards: "+str(record["rcards"])+"\n"+
                   "Minutes: "+str(record["minutes"])+"\n"+"\n"+
                   "Total National Caps: "+str(record["n-caps"])+"\n"+
                   "Total National Goals: "+str(record["n-goals"])+"\n")
                print(fact)

                panel_pie.config(text=fact,font=150,fg='white',image='')
################################################################################
```

Η συνάρτηση sort κατατάσσει τους ποδοσφαιριστές σύμφωνα με την επίδοση τους στο μέτρο που επιλέγει ο χρήστης, καλείται όταν ο χρήστης πατήσει το αντίστοιχο κουμπί αφού έχει πρώτα επιλέξει το μέτρο από την αντίστοιχη dropdown list. Η συνάρτηση αρχικά διαβάζει το μέτρο ως προς το οποίο θα γίνει η ταξινόμηση και δημιουργεί μια τριάδα που αποτελείται από το κλειδί, το όνομα, και το μέτρο που αντιστοιχεί στον κάθε ποδοσφαιριστή που ανήκει στον υποκύβο που έχει καθοριστεί. Στην συνέχεια ταξινομεί την τριάδα με βάση το μέτρο και καλεί την συνάρτηση top_percent, η οποία πρόκειται για μια απλή συνάρτηση που επιστρέφει για κάθε τιμή μιας λίστας το κορυφαίο Χ ποσοστό στο οποίο ανήκει. Τέλος προσθέτει στην listbox την νέα ταξινομημένη λίστα των ποδοσφαιριστών μαζί με το κορυφαίο ποσοστό στο οποίο ανήκει ο καθένας εξ αυτών. Τα επόμενα δύο αποσπάσματα αντιστοιχούν στην συνάρτηση sort και την βοηθητική συνάρτηση top_percent.

```
372 '''
373 Sort the players in cells according to measure
374 '''
375 ################################################################################
376 def sort():
377 
378     aggr = sort_var.get()
379     if aggr=='Age': val = 'age.age_ex'
        ...
388     elif aggr=='Total National Goals': val = 'n-goals'
389 
390     names=[]
391     values=[]
392     keys=[]
393 
394     a = cell_define()
395     result = browser.facts(cell=a)
396 
397     for record in result:
398         names.append(record["name"])
399         values.append(record[val])
400         keys.append(record["__fact_key__"])
401 
402     ### SORT RESULTS
403     tuples = sorted(zip(values, names, keys))
404     values, names, keys = [t[0] for t in tuples],[t[1] for t in tuples]
405                                                 ,[t[2] for t in tuples]
406 
407     if val != "age.age_ex" :
408         values, names, keys = values[::-1], names[::-1], keys[::-1]
409 
410     top=top_percent(values)
411 
420     i=0
421     listbox.insert(END, "\n")
422     listbox.insert(END, "\n Sorted Players List:")
423     while i<len(values):
424         rec=str(keys[i])+","+spaces[i]+str(i+1)+". "+names[i]+", "
425                            +aggr+": "+str(values[i])+", Top "+str(top[i])+"%"
426         print(rec)
427         listbox.insert(END, rec)
428         i+=1
429 
430 ################################################################################
```

```
347 '''
348 Return top X percent for every player in cell
349 '''
350 ################################################################################
351 def top_percent(sizes):
352 
353     perc=[]
354     i=0
355     while i < len(sizes):
356         j=i+1
357         count=0
358         while j < len(sizes):
359             if sizes[j] == sizes[i]:
360                 count+=1
361                 j+=1
362             else: break
363         percentage=round(((i+1+count)/len(sizes))*100,3)
364         perc.append(percentage)
365         i+=1
366 
367     return perc
368 
369 ################################################################################
```

Η συνάρτηση plot στην αρχή διαβάζει τις μεταβλητές που καθορίζουν το είδος του διαγράμματος που ζητείται, την συνάθροιση που ζητείται να αναπαρασταθεί, τον υποκύβο που καθορίζεται από το slicing & dicing και την διάσταση ως προς την οποία γίνετε, εάν έχει ζητηθεί, αναλυτική κάθοδος. Στην συνέχεια δημιουργούνται οι λίστες sizes και labels οι οποίες περιέχουν τα μεγέθη και τα αναγνωριστικά των στοιχείων που θα αναπαρασταθούν στο διάγραμμα. Οι πίτες, τα ραβδογράμματα και τα θηκογράμματα δημιουργούνται μέσα από την συνάρτηση plot με την χρήση των συναρτήσεων της βιβλιοθήκης matplotlib.pyplot, αντίθετα ο πίνακας στατιστικών δημιουργείται καλώντας την συνάρτηση stats_table η οποία δημιουργεί έναν πίνακα με όλες τις συναθροίσεις. Τέλος η συνάρτηση prepare που καλείται από την plot είναι μια βοηθητική συνάρτηση που προετοιμάζει τα δεδομένα για την καλύτερη εικόνα των διαγραμμάτων. Ακολουθεί ο κώδικας που αντιστοιχεί στην συνάρτηση prepare και στην συνέχεια αυτός που αντιστοιχεί στην συνάρτηση plot.

```
'''
Prepare Values for plotting
'''
################################################################################
def prepare(sizes,labels):

    ### REMOVING ZEROS
    k=0
    while k<len(labels):
        if sizes[k]==0:
            sizes.pop(k)
            labels.pop(k)
        else:
            k += 1

    ### SORT RESULTS
    tuples = sorted(zip(sizes, labels))
    sizes, labels = [t[0] for t in tuples], [t[1] for t in tuples]
    sizes, labels = sizes[::-1], labels[::-1]

    ### UNIFY AS "OTHERS" THOSE < 3%
    i = 0
    flag=1
    total=0
    while i < len(sizes):
        if (sizes[i]/sum(sizes))>0.03:
            i +=1
        else:
            if flag:
                cut_point=i
                flag=0
            total += sizes[i]
            i +=1
    if flag==0:
        sizes = sizes[0:cut_point]
        sizes.append(total)
        labels = labels[0:cut_point]
        labels.append('Others')

    return sizes,labels
################################################################################
```

```
124 '''
125 Creates plots (Pie, Barchart, Boxplot)
126 '''
127 ###############################################################################
128 def plot():
129     global img_pie
131     ### READ VARIABLES
132     dr = drill_var.get()
133     a = drilldown_define()
134     b = cell_define()
135     plot_type=plt_var.get()
137     if plot_type=="Stats Table":
138         stats_table(b)
139         return
141     aggr = aggr_var.get()
142     if aggr== 'Players Count': aggr = 'players'
        • • •
151     elif aggr=='Total National Goals': aggr = 'n-goals'
153     ### LABELS AND SIZES TO APPEAR IN PLOT
154     if a!=None:
155         result = browser.aggregate(cell=b,drilldown=a)
157         sizes=[]
158         labels=[]
159         for record in result:
160             sizes.append(record[aggr])
161             if dr=='National Team': labels.append(record["nat_team"])
                • • •
167             elif dr=='Role': labels.append(record["position.role"])
169         boxplot_sizes=sizes
170         sizes,labels=prepare(sizes,labels)
172     else:
173         result = browser.facts(cell=b)
174         boxplot_sizes=[]
175         for record in result:
176             if aggr == 'age.age_ex':
177                 boxplot_sizes.append(int(record[aggr]))
178             else:
179                 boxplot_sizes.append(record[aggr])
181     ### CREATE PLOT
182     fig = plt.figure()
183     fig.patch.set_facecolor('darkgreen')
184
185     if plot_type=='Pie':
186         plt.pie(sizes, labels=labels,autopct='%1.1f%%',
187                 shadow=True, startangle=0)
188         plt.axis('equal')
189     elif plot_type=='Bar Chart':
191         short_labels=[]
192         for label in labels:
193             short_labels.append(label[:8])
195         y_pos = np.arange(len(short_labels))
196         plt.barh(y_pos,sizes)
197         plt.yticks(y_pos, short_labels)
198     elif plot_type=='Boxplot':
199         plt.boxplot(boxplot_sizes)
200
201     plt.savefig('pie.png', facecolor=fig.get_facecolor())
202     plt.show()
203     img_pie = PhotoImage(file="pie.png")
205     panel_pie.config(image=img_pie)
207 ###############################################################################
```

4.6 Οδηγίες λήψης και εκτέλεσης της εφαρμογής

Ο κώδικας που παρουσιάστηκε προηγουμένως, η βάση δεδομένων, το λογικό μοντέλο, τα απαραίτητα αρχεία για την αρχικοποίηση του Cubes καθώς και οτιδήποτε άλλο χρειάζεται για την εκτέλεση της εφαρμογής βρίσκονται στο Github και μπορούν να ληφθούν από τον ακόλουθο σύνδεσμο:

https://github.com/petje93/World-Cup-2018-OLAP

Αναλυτικά τα αρχεία που βρίσκονται στον σύνδεσμο φαίνονται στον ακόλουθο πίνακα:

Όνομα	Περιγραφή
WC18.csv	Βάση Δεδομένων σε csv
preparation.py	Δημιουργία αρχείου SQLite από csv
slicer.ini	Αρχείο αρχικοποίησης του Slicer Server
world_cup.sqlite	Βάση Δεδομένων σε SQLite (Αυτή που χρησιμοποιεί η εφαρμογή)
world_cup_model.json	Λογικό Μοντέλο
world_cup_olap.py	Εφαρμογή Python

Πίνακας 4.2: Αρχεία Εφαρμογής

Για την εκτέλεση της εφαρμογής θα πρέπει προηγουμένως να έχει ληφθεί το Cubes όπως εξηγήθηκε στο 3ο Κεφάλαιο, καθώς και να υπάρχουν στον υπολογιστή του χρήστη όλες οι βιβλιοθήκες που χρησιμοποιήθηκαν στην εφαρμογή δηλαδή οι tkinter, matplotlib, numpy, pandas καθώς και η βιβλιοθήκη sqlalchemy πάνω στην οποία βασίζεται το εσωτερικό μέρος SQL του Cubes.

Για την εκτέλεση της εφαρμογής μεταβαίνετε στον φάκελο που λήφθηκε από το Github μέσω της γραμμής εντολών και δίνετε την εντολή

python world_cup_olap.py

Για την εκτέλεση του εξυπηρετητή Slicer δίνετε την εντολή

slicer serve slicer.ini

Σε περίπτωση που επιθυμείται η προσθήκη νέων στοιχείων στην βάση δεδομένων μέσω του αρχείου csv, πρωτού εκτελεστεί η εφαρμογή ή ο εξυπηρετητής θα πρέπει να εκτελεστεί το αρχείο preparation.py έτσι ώστε να ανανεωθεί το αρχείο sqlite δίνοντας την εντολή

python preparation.py

Κεφάλαιο 5

Συμπεράσματα Ανάλυσης Δεδομένων

Στο κεφάλαιο αυτό θα παρουσιαστούν ορισμένα συμπεράσματα που εξήχθησαν με την χρήση της πλατφόρμας που παρουσιάστηκε στο προηγούμενο κεφάλαιο για την ανάλυση των δεδομένων για το Παγκόσμιο Κύπελλο του 2018. Αρχικά θα παρουσιαστούν ορισμένα στατιστικά στοιχεία και διαγράμματα που προκύπτουν από ολόκληρο τον κύβο ενώ στην συνέχεια θα γίνει εφαρμογή αναλυτικών καθόδων στις διαστάσεις του κύβου.

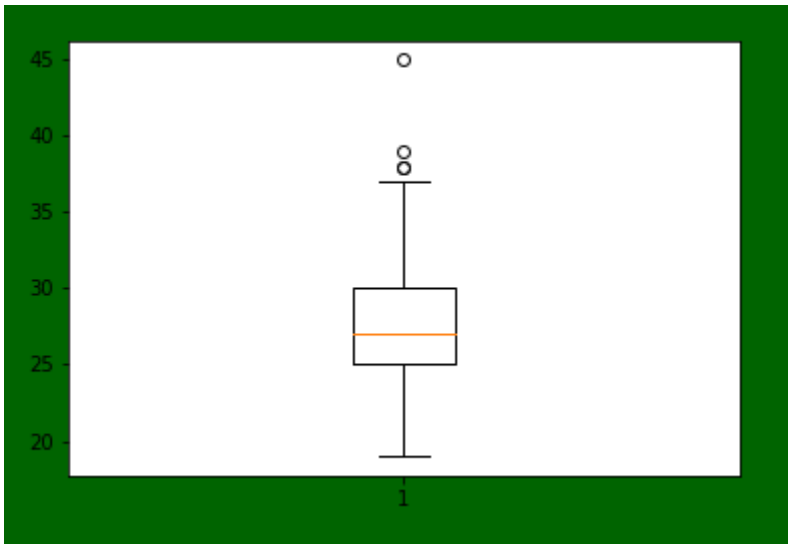
5.1 Γενικά Συμπεράσματα

Αρχικά καλώντας την συνάρτηση Aggregate σε ολόκληρο τον κύβο παρατηρείται πως από τους 736 ποδοσφαιριστές που στελέχωσαν τις ομάδες αγωνίστηκαν οι 604 (82%), ενώ συνολικά στην διοργάνωση επιτεύχθηκαν 157 τέρματα από 110 διαφορετικούς παίκτες. Ακολουθεί ο πίνακας στατιστικών ολόκληρου του κύβου. Ιδιαίτερη αίσθηση προκαλεί το γεγονός πως σε ολόκληρη την διοργάνωση υπήρξαν μόλις 4 αποβολές.

	Total	Max	Average
Goals	157.0	6.0	0.213
Assists	125.0	3.0	0.17
Caps	1790.0	7.0	2.432
Minutes	129846.0	694.0	176.421
Yellow Cards	219.0	3.0	0.298
Red Cards	4.0	1.0	0.005

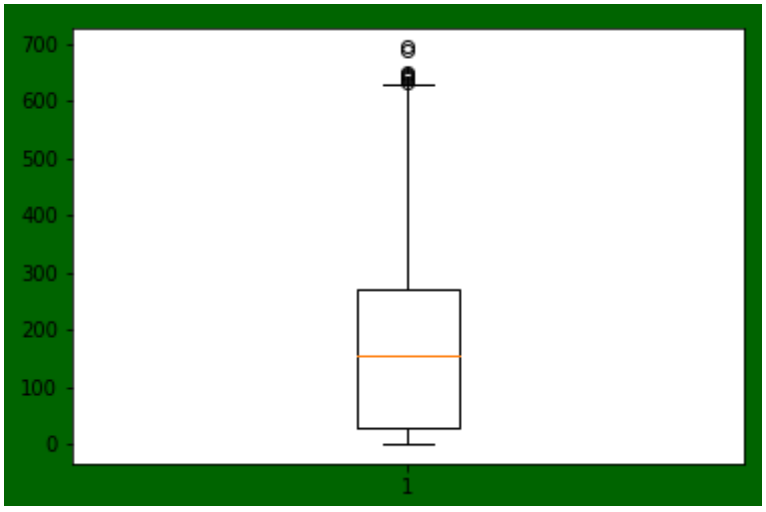
Σχήμα 5.1: Στατιστικά του συνόλου των ποδοσφαιριστών

Από το θηκόγραμμα των ηλικιών των ποδοσφαιριστών παρατηρείται πως η μέση ηλικία των ποδοσφαιριστών που έλαβαν μέρος στην διοργάνωση κυμαίνεται μεταξύ των 26 και 27. Περαιτέρω φαίνεται πως υπάρχουν 4 ακραία σημεία στο σύνολο των ποδοσφαιριστών που ταξινομώντας κατά ηλικία διακρίνεται πως πρόκειται για τους Essam El-Hadary (45 ετών), Rafael Marquez (39 ετών), Tim Cahill (38 ετών) και Sergei Ignashevich (38 ετών). Αντιθέτως παρατηρείται ότι δεν υπάρχουν ακραίες τιμές νεαρών ποδοσφαιριστών.



Σχήμα 5.2: Θηκόγραμμα ηλικιών των ποδοσφαιριστών

Στην συνέχεια ακολουθεί το θηκόγραμμα του συνολικού χρόνου συμμετοχής κάθε ποδοσφαιριστή στην διοργάνωσης. Όπως παρατηρήθηκε και στον πίνακα στατιστικών παραπάνω ο μέσος χρόνος συμμετοχής είναι τα 176 λεπτά, ενώ παρατηρούνται ορισμένες ακραίες τιμές ποδοσφαιριστών που έχουν αγωνιστεί περισσότερα από 600 λεπτά.



Σχήμα 5.3: Θηκόγραμμα χρόνου συμμετοχής των ποδοσφαιριστών

Για να εντοπιστούν οι ποδοσφαιριστές στους οποίους αντιστοιχούν οι ακραίες τιμές γίνεται ταξινόμηση κατά τον χρόνο συμμετοχής. Παρατηρείται πως υπάρχουν 13 ποδοσφαιριστές με περισσότερα από 600 λεπτά συμμετοχής και όλοι αγωνίζονται σε ομάδες που έφτασαν μέχρι τα ημιτελικά και συνεπώς έπαιξαν 7 αγώνες στην διοργάνωση. Οι 8 από τους 13 αγωνίζονται στην εθνική ομάδα της Κροατίας γεγονός που είναι λογικό καθώς η συγκεκριμένη ομάδα έφτασε στην παράταση σε τρεις αγώνες ενώ επίσης βασιζόταν σε ένα μικρότερο σύνολο βασικών ποδοσφαιριστών. Πρώτος στην συγκεκριμένη λίστα είναι ο Luka Modric ο οποίος αναδείχθηκε και πολυτιμότερος παίκτης της διοργάνωσης.

#	Name	Team	Minutes	Top X %
1	Luka Modric	Croatia	694	0.136%
2	Jordan Pickford	England	690	0.272%
3	Dejan Lovren	Croatia	650	0.408%
4	John Stones	England	645	0.543%
5	Harry Maguire	England	644	0.679%
6	Ivan Rakitic	Croatia	638	0.815%
7	Ivan Perisic	Croatia	632	0.951%
8	Thibaut Courtois	France	630	1.495%
9	Raphael Varane	France	630	1.495%
10	Domagoj Vida	Croatia	630	1.495%
11	Danijel Subasic	Croatia	630	1.495%
12	Mario Mandzukic	Croatia	609	1.630%
13	Sime Vrsaljko	Croatia	607	0.766%

Πίνακας 5.1: Κορυφαίοι Παίκτες σε Χρόνο Συμμετοχής

5.2 Αναλυτική Κάθοδος κατά Εθνική Ομάδα

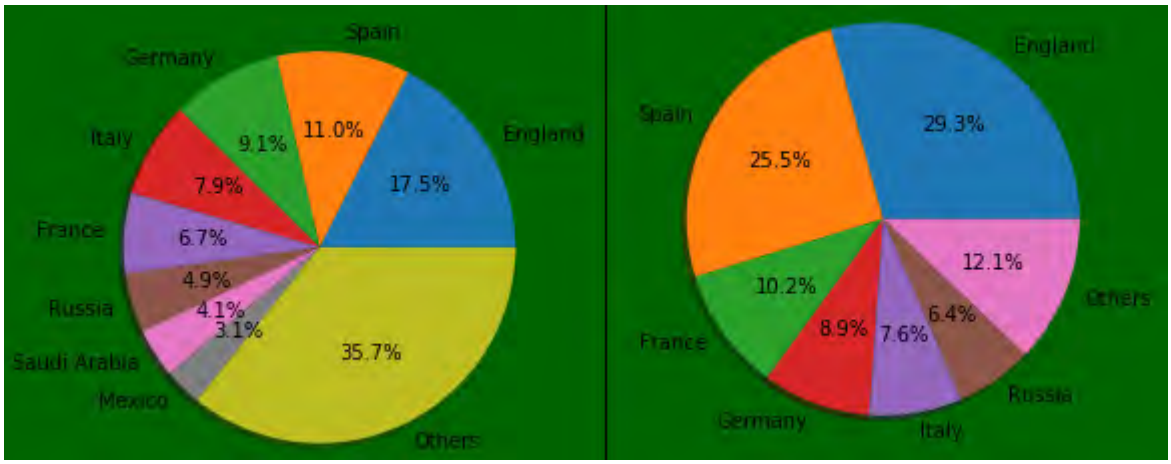
Στον παρακάτω πίνακα παρουσιάζονται οι συναθροίσεις εάν επιλέξουμε αναλυτική κάθοδο στην διάσταση της εθνικής ομάδας, δηλαδή μπορούμε να δούμε τις επιδόσεις των ποδοσφαιριστών ανά την εθνική ομάδα στην οποία αγωνίζονται. Παρατηρούμε πως οι τέσσερις ομάδες που έφτασαν ως τα ημιτελικά πέτυχαν όπως ήταν αναμενόμενο και τα περισσότερα τέρματα. Στις περιπτώσεις της Σουηδίας και της Κόστα Ρίκα παρατηρούνται περισσότερες τελικές πάσες από ότι γκολ, αυτό συμβαίνει καθώς έχουν χρεωθεί τελικές πάσες σε αυτογκόλ. Η μοναδική ομάδα που χρησιμοποίησε όλους τους ποδοσφαιριστές της είναι η Τυνησία ενώ αυτή που χρησιμοποίησε τους λιγότερους είναι η Αυστραλία (15/23). Τέλος το Βέλγιο ήταν η μοναδική ομάδα που είχε διψήφιο αριθμό ποδοσφαιριστών που σκόραραν στην διοργάνωση.

Team	Goals	Assists	Y-Cards	R-Cards	Scored	Played
Argentina	6	6	11	0	5/23	20/23
Australia	2	0	7	0	1/23	15/23
Belgium	15	13	11	0	10/23	21/23
Brazil	8	7	7	0	6/23	18/23
Colombia	6	5	9	1	4/23	20/23
Costa Rica	1	2	6	0	1/23	20/23
Croatia	13	9	15	0	8/23	21/23
Denmark	3	3	6	0	3/23	20/23
Egypt	2	1	5	0	1/23	16/23
England	12	7	8	0	6/23	21/23
France	12	9	12	0	6/23	21/23
Germany	2	2	2	1	2/23	20/23
Iceland	2	1	3	0	2/23	18/23
Iran	1	1	7	0	1/23	16/23
Japan	6	5	5	0	5/23	18/23
Mexico	3	2	9	0	3/23	19/23
Morocco	2	1	8	0	2/23	19/23
Nigeria	3	3	4	0	2/23	16/23
Panama	1	2	11	0	1/23	20/23
Peru	2	1	5	0	2/23	17/23
Poland	2	2	3	0	2/23	21/23
Portugal	6	4	7	0	3/23	18/23
Russia	10	9	6	1	5/23	19/23
Saudi Arabia	2	2	1	0	2/23	20/23
Senegal	3	3	6	0	3/23	18/23
Serbia	2	1	9	0	2/23	18/23
South Korea	3	2	10	0	2/23	19/23
Spain	6	3	2	0	4/23	17/23
Sweden	5	6	8	0	4/23	19/23
Switzerland	5	4	9	1	5/23	17/23
Tunisia	5	5	4	0	4/23	23/23
Uruguay	6	4	3	0	3/23	19/23

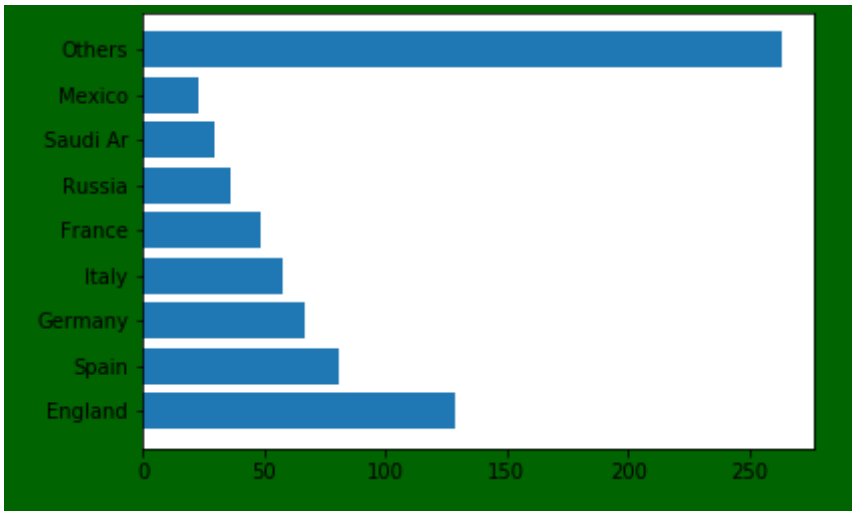
Πίνακας 5.2: Συνολικές επιδόσεις ποδοσφαιριστών ανά εθνική ομάδα

5.3 Αναλυτική Κάθοδος κατά Πρωτάθλημα

Εκτελώντας αναλυτική κάθοδο κατά τα πρωταθλήματα στα οποία αγωνίζονται οι ποδοσφαιριστές σε συλλογικό επίπεδο συμπεραίνουμε πως το 52% των ποδοσφαιριστών αγωνίζονται σε κάποιο από τα πέντε ισχυρότερα πρωταθλήματα της Ευρώπης, το αποκαλούμενο Big 5, δηλαδή τα πρωταθλήματα της Αγγλίας, της Ισπανίας, της Γερμανίας, της Ιταλίας και της Γαλλίας. Όσον αφορά τα τέρματα που επιτεύχθηκαν στην διοργάνωση, αυτά που είχαν ως σκόρερ ποδοσφαιριστή που αγωνίζεται σε σύλλογο του Big 5 αποτελούν το 81% του συνόλου των τερμάτων. Τόσο στο σύνολο των ποδοσφαιριστών όσο και στα τέρματα το πρωτάθλημα που έπεται τα πέντε κορυφαία είναι το Ρωσικό. Τα παραπάνω αποτυπώνονται στα διαγράμματα πίτας που ακολουθούν. Το πλήθος των ποδοσφαιριστών ανά πρωτάθλημα φαίνεται στο ραβδόγραμμα ακριβώς κάτω από τις πίτες.



Σχήμα 5.4: Πλήθος ποδοσφαιριστών (αριστερά) και πλήθος τερμάτων (δεξιά) ανά πρωτάθλημα



Σχήμα 5.5: Ραβδόγραμμα πλήθους ποδοσφαιριστών ανά πρωτάθλημα

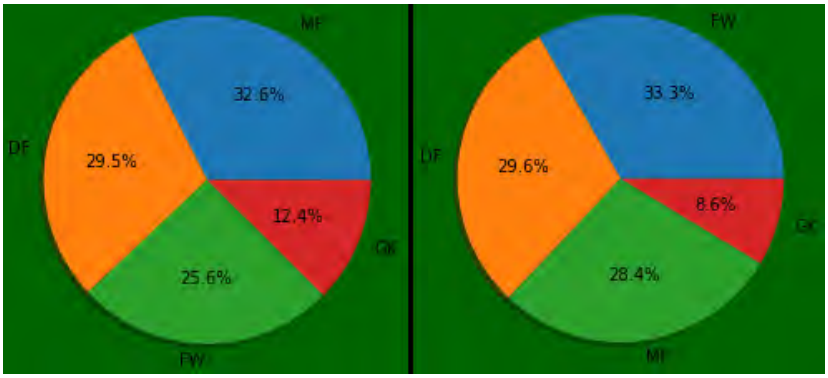
Στην συνέχεια πραγματοποιείται μια σύγκριση μεταξύ των ποδοσφαιριστών που αγωνίζονται στα δύο μεγαλύτερα πρωταθλήματα, στην επόμενη εικόνα φαίνονται τα στατιστικά των ποδοσφαιριστών που αγωνίζονται στην Αγγλία (επάνω) και στην Ισπανία (κάτω). Από το πρωτάθλημα της Αγγλίας αγωνίστηκαν οι 115 από τους 129 ποδοσφαιριστές που βρισκόταν στις αποστολές των ομάδων, ενώ από αυτό της Ισπανία αγωνίστηκαν οι 75 από τους 81.

	Total	Max	Average
Goals	46.0	6.0	0.357
Assists	29.0	2.0	0.225
Caps	414.0	7.0	3.209
Minutes	31748.0	690.0	246.109
Yellow Cards	46.0	3.0	0.357
Red Cards	0.0	0.0	0.0

	Total	Max	Average
Goals	40.0	4.0	0.494
Assists	25.0	3.0	0.309
Caps	267.0	7.0	3.296
Minutes	19989.0	694.0	246.778
Yellow Cards	34.0	2.0	0.42
Red Cards	1.0	1.0	0.012

Σχήμα 5.6: Σύγκριση μεταξύ στατιστικών ποδοσφαιριστών που αγωνίζονται στα πρωταθλήματα Αγγλίας (επάνω) και Ισπανίας (κάτω)

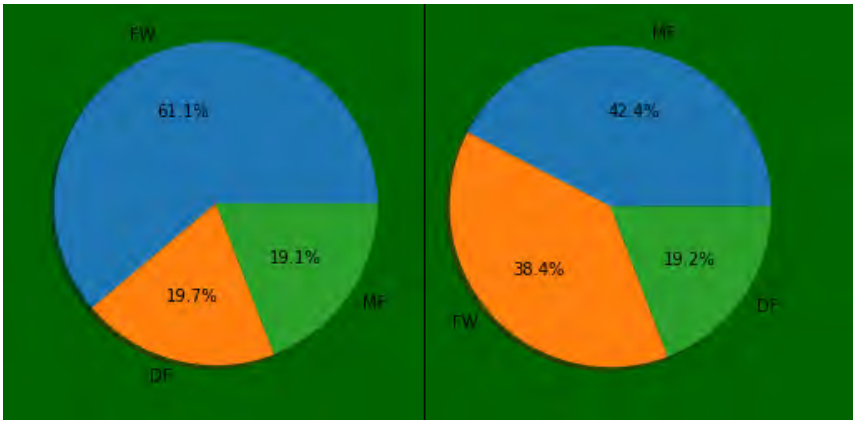
Οι πολύ υψηλές τιμές που παρατηρούνται σε γκολ και τελικές πάσες από τους ποδοσφαιριστές του πρωταθλήματος της Ισπανίας δείχνουν ότι πιθανόν μεγαλύτερο ποσοστό αυτών αγωνίζεται στην επίθεση σε σχέση με αυτούς που αγωνίζονται στο Αγγλικό πρωτάθλημα. Αυτό επιβεβαιώνεται στην παρακάτω εικόνα με αναλυτική κάθοδο κατά θέση και τομή στο πρωτάθλημα Αγγλίας (αριστερά) και Ισπανίας (δεξιά).



Σχήμα 5.7: Ποδοσφαιριστές που αγωνίζονται στα πρωταθλήματα Αγγλίας και Ισπανίας ανά θέση

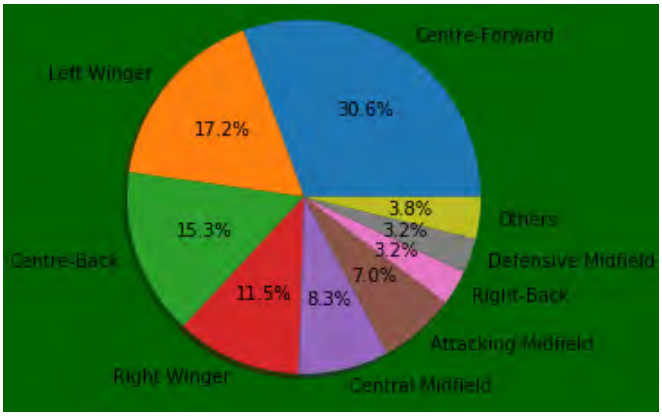
5.4 Αναλυτική Κάθοδος κατά Θέση

Όσον αφορά την θέσης που αγωνίζονται οι ποδοσφαιριστές που σκόραραν στην διοργάνωση, αυτό παρουσιάζεται στο πρώτο διάγραμμα της επόμενης εικόνας που προκύπτει μετά από αναλυτική κάθοδο ενός επιπέδου κατά την διάσταση της θέσης. Στο διάγραμμα αυτό όπως ήταν αναμενόμενο φαίνεται πως οι επιθετικοί αποτελούν το συντριπτικά μεγαλύτερο ποσοστό ενώ αξίζει να σημειωθεί πως οι παίκτες της μεσαίας γραμμής δεν πέτυχαν πολλά τέρματα στην διοργάνωση καθώς βρίσκονται πίσω από τους παίκτες της άμυνας. Ωστόσο συμμετείχαν σε μεγάλο βαθμό στην δημιουργία των τερμάτων γεγονός που αντικατοπτρίζεται στο δεύτερο διάγραμμα που δείχνει το ποσοστό των συνολικών τελικών πασών (assist) ανά θέση, στο οποίο οι παίκτες της μεσαίας γραμμής σημειώνουν το υψηλότερο ποσοστό.



Σχήμα 5.8: Πλήθος τερμάτων και τελικών πασών ανά θέση

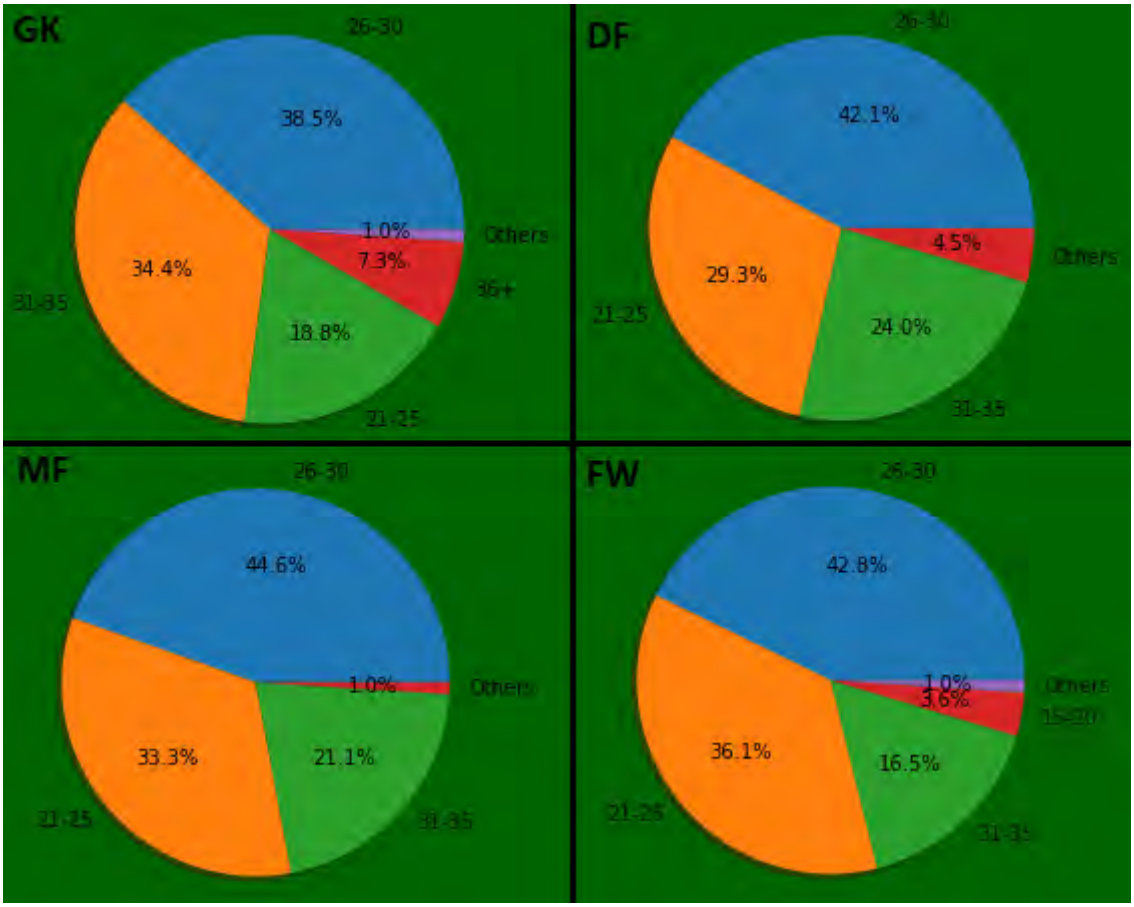
Άμα στο πρώτο διάγραμμα της προηγούμενης εικόνας εκτελέσουμε αναλυτική κάθοδο κατά ένα ακόμα επίπεδο δηλαδή κατά τον συγκεκριμένο ρόλο των ποδοσφαιριστών και όχι την γενική θέση τότε θα λάβουμε το αποτέλεσμα που φαίνεται στην επόμενη εικόνα. Σε αυτήν παρατηρούμαι το πολύ υψηλό ποσοστό τερμάτων που έχουν σκοράρει κεντρικοί αμυντικοί (15,3%) από το οποίο μπορούμε να εξάγουμε το συμπέρασμα πως σημειώθηκαν αρκετά τέρματα μετά από την εκτέλεση στημένων φάσεων καθώς είναι ο συνηθέστερος τρόπος που σκοράρει ένας κεντρικός αμυντικός.



Σχήμα 5.9: Πλήθος τερμάτων ανά ρόλο ποδοσφαιριστή

5.5 Αναλυτική Κάθοδος κατά Ηλικία

Στην παρακάτω εικόνα έχει πραγματοποιηθεί αναλυτική κάθοδο κατά ηλικία με τομή για κάθε μια θέση. Όπως φαίνεται σε όλες τις θέσεις το μεγαλύτερο ποσοστό αντιστοιχεί στην ηλικιακή ομάδα 26 έως 30, το οποίο είναι αναμενόμενο καθώς πρόκειται για το ηλικιακό εύρος κατά το οποίο οι ποδοσφαιριστές φτάνουν συνήθως στο αποκορύφωμα της απόδοσης τους. Για όλες τις θέσεις εκτός από αυτή του τερματοφύλακα το δεύτερο μεγαλύτερο ποσοστό αντιστοιχεί στο ηλικιακό εύρος 21 έως 25 και το τρίτο στο 31 έως 35. Στην θέση του τερματοφύλακα συμβαίνει το αντίστροφο, πράγμα που επιβεβαιώνει το γεγονός ότι οι τερματοφύλακες πραγματοποιούν τις καλύτερες χρονιές τους σε μεγαλύτερες ηλικίες από ότι οι υπόλοιποι ποδοσφαιριστές, επίσης αξίζει να σημειωθεί πως παρατηρείται και ένα σχετικά υψηλό ποσοστό (7,3%) για την ηλικιακή κατηγορία άνω των 36 για τους τερματοφύλακες. Τέλος οι νεαροί ποδοσφαιριστές του εύρους 15 έως 20 σημειώνουν το υψηλότερο ποσοστό τους (3,6%) στην θέση της επίθεσης.



Σχήμα 5.10: Πλήθος ποδοσφαιριστών κάθε θέσης ανά ηλικιακή ομάδα

Κεφάλαιο 6

Επίλογος

Παρακάτω θα ακολουθήσει μια σύνοψη των σημαντικότερων θεμάτων που αναλύθηκαν στην διπλωματική, των βασικότερων βημάτων που ακολουθήθηκαν για την δημιουργία της εφαρμογής και των κύριων συμπερασμάτων που προέκυψαν. Επίσης θα συζητηθούν ορισμένες ιδέες για μελλοντικές επεκτάσεις της εργασίας.

6.1 Σύνοψη και συμπεράσματα

Με την ραγδαία αύξηση του όγκου των δεδομένων την τελευταία δεκαετία, και με την ταυτόχρονη ανάπτυξη των μεθόδων συλλογής δεδομένων όλο και περισσότεροι τομείς προσπαθούν να επωφεληθούν χρησιμοποιώντας την γνώση που μπορεί να αντληθεί μέσα από τα δεδομένα για την λήψη αποφάσεων. Στο δεύτερο κεφάλαιο της εργασίας αυτής περιγράφηκαν οι τρόποι με τους οποίους μπορούν να επωφεληθούν οι φορείς που εμπλέκονται με τον αθλητισμό μέσα από την χρήση τεχνικών της επιστήμης των δεδομένων, ενώ παράλληλα επεξηγήθηκαν οι βασικές έννοιες που συνδέονται με μια αποθήκη δεδομένων.

Σκοπός της παρούσας διπλωματικής εργασίας ήταν η δημιουργία μιας εφαρμογής η οποία θα επιτύγχανε την άμεση αναλυτική επεξεργασία δεδομένων αθλητικών αγωνισμάτων. Για τον λόγο αυτό χρησιμοποιήθηκε το εργαλείο Cubes το οποίο επιτρέπει την επίτευξη της ζητούμενης λειτουργικότητας σε προγραμματιστικό περιβάλλον Python. Στο τρίτο κεφάλαιο παρουσιάστηκαν τα βασικά μέρη της αρχιτεκτονικής του συγκεκριμένου εργαλείου τα οποία συνοψίζονται παρακάτω:

- Logical Model : Μοντελοποίηση των δεδομένων ως πολυδιάστατοι κύβοι δεδομένων
- Aggregation Browser : Υπεύθυνος για την εκτέλεση των λειτουργιών άμεσης αναλυτικής επεξεργασίας, δηλαδή υπολογισμός συναθροίσεων, τεμαχισμός και κομμάτιασμα, αναλυτική κάθοδος
- Server : HTTP Server που καλύπτει τις βασικότερες δυνατότητες του Cubes
- Backends : Υπεύθυνα για την σύνδεση και την εργασία με τα φυσικά δεδομένα

Για τις ανάγκες της εργασίας δημιουργήθηκε μια βάση δεδομένων που περιλαμβάνει στοιχεία για όλους τους ποδοσφαιριστές που συμμετείχαν στο Παγκόσμιο Κύπελλο ποδοσφαίρου του 2018. Η βάση αυτή περιλαμβάνει 736 γραμμές και 14 στήλες, μέγεθος δηλαδή ιδιαίτερα μικρό σε σχέση με τις αποθήκες δεδομένων μεγάλων οργανισμών, παρ' όλα αυτά κρίθηκε κατάλληλη για να χρησιμοποιηθεί για την ανάλυση των δεδομένων της εργασίας αυτής που έχει ως στόχο να δείξει τον τρόπο με τον οποίο πραγματοποιείται η άμεση αναλυτική επεξεργασία. Η βάση δεδομένων αυτή παρουσιάζεται στο υποκεφάλαιο 4.1 της εργασίας ενώ στην συνέχεια εξηγείται ο τρόπος με τον οποίο μοντελοποιείται ως υπέρ-κύβος με τέσσερις διαστάσεις και έξι μέτρα από τα οποία προκύπτει πληθώρα συναθροίσεων.

Η εφαρμογή που δημιουργήθηκε περιγράφεται στο υποκεφάλαιο 4.5, μέσα από αυτή ο χρήστης μπορεί να καθορίσει τον υποκύβο, δηλαδή το υποσύνολο των δεδομένων, για το οποίο ενδιαφέρεται αρχικοποιώντας τις κατάλληλες παραμέτρους που επιτυγχάνουν τον τεμαχισμό και το κομμάτιασμα του κύβου, να επιλέξει τις συναθροίσεις που θέλει καθώς και να ζητήσει αναλυτική κάθοδο κατά κάποια διάσταση. Στην συνέχεια μεταβαίνει στο παράθυρο των αποτελεσμάτων στο οποίο παρέχονται διάφορα εργαλεία για την οπτικοποίηση του αποτελέσματος μέσω διαγραμμάτων καθώς επίσης και η δυνατότητα ταξινόμησης των ποδοσφαιριστών που συμπεριλαμβάνονται στο αποτέλεσμα και τέλος η προβολή λεπτομερειών για τον καθένα. Ο κώδικας που υλοποιεί την συγκεκριμένη εφαρμογή επεξηγείται ενδελεχώς στο υποκεφάλαιο 4.5.2

Τέλος στο κεφάλαιο 5 παρουσιάζονται κάποια βασικά συμπεράσματα που μπορούν να προκύψουν από την άμεση αναλυτική επεξεργασία της βάσης δεδομένων του Παγκοσμίου Κυπέλλου του 2018 με την χρήση της εφαρμογής που υλοποιήθηκε. Έγινε προσπάθεια να καλυφθούν ορισμένα από τα βασικότερα ερωτήματα που αφορούν την συγκεκριμένη βάση ως προς τις επιδόσεις των ποδοσφαιριστών ενώ επίσης καλύφθηκαν παραδείγματα ερωτημάτων με αναλυτική κάθοδο για όλες τις διαστάσεις που έχουν οριστεί. Προφανώς και μέσω της εφαρμογής μπορούν να εξαχθούν και πιο εξειδικευμένα συμπεράσματα τα οποία εξαρτώνται από τα ακριβή ζητούμενα του χρήστη.

6.2 Μελλοντικές επεκτάσεις

Η κύρια κατεύθυνση κατά την οποία μπορεί να επεκταθεί η εφαρμογή που δημιουργήθηκε στην παρούσα διπλωματική εργασία είναι η σύνδεση της με μια πολύ μεγαλύτερη αποθήκη δεδομένων. Θα μπορούσαν να προστεθούν δεδομένα τόσο από προηγούμενες διοργανώσεις του παγκοσμίου κυπέλλου έτσι ώστε να μπορούν να προκύψουν πλούσια συμπεράσματα για την εξέλιξη της διοργάνωσης και του αθλήματος γενικότερα. Περαιτέρω μπορούν να προστεθούν δεδομένα από διαφορετικές ποδοσφαιρικές διοργανώσεις όπως για παράδειγμα το κύπελλο πρωταθλητριών αλλά και τα μεγαλύτερα εθνικά πρωταθλήματα, με αυτόν τον τρόπο μπορούν να προκύψουν συμπεράσματα για τα ιδιαίτερα χαρακτηριστικά της κάθε διοργάνωσης και να επισημανθούν οι διαφορές και οι ομοιότητες στον τρόπο που προσεγγίζεται το άθλημα σε αυτές.

Περαιτέρω θα μπορούσαν να προστεθούν και άλλοι κύβοι πέραν από αυτόν που χρησιμοποιείται στην διπλωματική και σαν ελάχιστο στοιχείο έχει έναν ποδοσφαιριστή. Θα ήταν δυνατό να

δημιουργηθούν, εφόσον υπήρχαν τα κατάλληλα δεδομένα, κύβοι που θα είχαν ως πυρήνα μια ομάδα, έναν προπονητή, έναν αγώνα της διοργάνωσης ή και ένα γκολ επιτρέποντας κατά αυτόν τον τρόπο την βαθύτερη ανάλυση του αγωνίσματος και την εξαγωγή περισσότερο πολύπλοκων συμπερασμάτων.

Επίσης στο κομμάτι της οπτικοποίησης των αποτελεσμάτων που προκύπτουν από την άμεση αναλυτική επεξεργασία θα μπορούσαν να προστεθούν περισσότερες δυνατότητες αναπαράστασης μέσω της παραγωγής διαγραμμάτων διαφορετικών τύπων. Τέλος μπορεί να προστεθεί μια νέα λειτουργία που θα επιτρέπει την σύγκριση μεταξύ των δεδομένων που επιθυμεί ο χρήστης έτσι ώστε να μπορεί να τα βλέπει ταυτόχρονα και να τα εξετάσει αναλυτικότερα για να εντοπίσει τις διαφορές και τις ομοιότητες που παρουσιάζουν.

Βιβλιογραφία

[1] E. Bernier, Y. Bédard, T. Badard και F. Hubert. *UMapIT© (Unrestricted Mapping Interactive Tool): Merging the datacube paradigm with an occurrence-based approach to support on-demand web mapping*, σελίδες 187–204. 2007.

[2] E. Burns. Analytics in sports gets growing front-office role. `https://searchbusinessanalytics.techtarget.com/feature/Analytics-in-sports-gets-growing-front-office-role`, 2014. Ημερομηνία πρόσβασης: 18-11-2018.

[3] M. Loukides. What is data science? `https://www.oreilly.com/ideas/what-is-data-science`, 2010. Ημερομηνία πρόσβασης: 15-8-2018.

[4] T. Macaulay. How data analytics is transforming sports on and off the pitch. `https://www.techworld.com/data/how-data-analytics-is-transforming-sports-on-off-pitch-3668759/`, 2017. Ημερομηνία πρόσβασης: 20-11-2018.

[5] J. Manyika, M. Chui, B. Brown, J. Bughin, R. Dobbs, C. Roxburgh και A. Hung Byers. *Big data: The next frontier for innovation, competition, and productivity*. McKinsey, 2011.

[6] J. M. Perez Martinez, R. Berlanga, M. J. Aramburu και T. B. Pedersen. Integrating Data Warehouses with Web Data: A Survey. *IEEE Transactions on Knowledge and Data Engineering*, 20(7):940–955, 2008.

[7] F. Shahzeb. How Data Analytics Helps Coaches in Planning. `https://www.workinsports.com/blog/how-data-analytics-helps-coaches-in-planning/`, 2017. Ημερομηνία πρόσβασης: 15-11-2018.

[8] L. Steinberg. CHANGING THE GAME: The Rise of Sports Analytics. `https://www.forbes.com/sites/leighsteinberg/2015/08/18/changing-the-game-the-rise-of-sports-analytics/#55ad18014c1f`, 2015. Ημερομηνία πρόσβασης: 15-11-2018.

[9] P. Tan, M. Steinbach και V. Kumar. *Introduction to Data Mining*. Pearson, 2005.

[10] Transfermarkt, Πηγή Δεδομένων. `https://www.transfermarkt.com`. Ημερομηνία πρόσβασης: 10-8-2018.

[11] S. Urbanek. Cubes - Lightweight Python OLAP. EuroPython, 2012.

[12] S. Urbanek. Cubes Documentation: Release 1.1. `https://cubes.readthedocs.io/en/v1.1/`, 2016. Ημερομηνία πρόσβασης: 20-6-2018.

[13] K. Wendt. *Traffic Monitor: Data Display for Traffic Visualisation at Airports*. Διδακτορική Διατριβή, 2007.

[14] R. Wrembel και C. Koncilia. *Data Warehouses And Olap: Concepts, Architectures And Solutions*. IRM Press, Hershey, PA, United States, 2006.

Συντομογραφίες

κ.λπ.	και λοιπά
κ.ο.κ	και ούτω καθεξής
π.χ.	παραδείγματος χάριν
OLAP	Online Analytical Processing
GUI	Graphical User Interface
JSON	JavaScript Object Notation
HTTP	Hypertext Transfer Protocol
BI	Business Intelligence
SQL	Structured Query Language
csv	Comma-seperated values
DBMS	Database Management System
ORM	Object-Relational Mapper

Ορολογία - Γλωσσάρι

Ελληνικός όρος	Αγγλικός όρος
άμεση αναλυτική επεξεργασία	online analytical processing
επιχειρησιακή νοημοσύνη	business intelligence
αποθήκη δεδομένων	data warehouse
εφαρμογές αναφοράς δεδομένων	reporting applications
μεταδεδομένα	metadata
ερώτημα	query
αναλυτική κάθοδος	drilldown
τεμαχισμός	slicing
κομμάτιασμα	dicing
κάνναβος	grid
κελί	cell
τομή	cut
μέτρο	measure
πίνακας γεγονότων	fact table
γραφικό περιβάλλον χρήστη	graphical user interface
παράθυρο	window
συνάθροιση	aggregation
σύνδεση	join
σετ δεδομένων	dataset
σχήμα αστέρα	star schema
σχήμα χιονονιφάδας	snowflake schema
στατιστικές αναλύσεις	analytics
εξυπηρετητής	server
εσωτερικό τμήμα	backend
περιηγητής συναθροίσεων	aggregation browser
εργασιακός χώρος	workspace
διασταυρωμένη πινακοποίηση	cross table
τελική πάσα	assist