LABORATORY OF DATA SCIENCE

PROGETTO 2021/2022

Andrea Carnevale

Federico Canepuzzi

Data Science and Business Informatics

Università di Pisa

Anno accademico 2021/2022

Contents

Parte 1

1.1 Preprocessing

Per questa prima parte del progetto sono stati messi a disposizione 4 file CSV: 'tennis', 'male_players', 'female_players' e 'countries'. Il file principale è 'tennis' che contiene la maggior parte delle informazioni da inserire nel Data Warehouse; per questo la fase di preprocessing si è concentrata principalmente su questo file. Questa fase è stata svolta su un Notebook in modo da favorire la visualizzazione dei risultati e rendere facilmente comprensibili le varie operazioni. È stata sfruttata principalmente la libreria Pandas per lavorare su dataframe. Alla fine di questo task, i nuovi file tennis, male_players e female_players sono stati salvati ancora nella cartella 'data' ma con l'aggiunta di '_new' al nome; questi file saranno poi utilizzati per l'assignment 1. Il nuovo file countries è invece stato direttamente salvato nella cartella 'table' visto che sarà utilizzato solo per l'assignment 2. In maniera schematica vengono qui riportate le varie operazioni di preprocessing eseguite sui 4 file CSV.

1.1.1 Tennis.csv

- 'Duplicate rows': 309 rows sono state eliminate perché presentavano gli stessi valori su tutti gli attributi in un altro record.
- 'Match errors': Erano presenti 6 record con una partita tra giocatori con lo stesso id e lo stesso nome. Sono stati eliminati questi 6 record.
- 'Player_entry': Gli attributi 'winner_entry' e 'loser_entry' sono stati eliminati perché non utilizzati nel DW.
- 'Player_Id': 49 id diversi presentavano per lo stesso 'player_id' due nomi diversi. Stampando i nomi è stato osservato che 2 id presentavano nomi diversi per errori di spaziatura e/o digitalizzazione, mentre gli altri 47 facevano riferimento a persone distinte. È stato perciò necessario creare per ogni coppia con lo stesso id, un nuovo id che è stato associato ad uno dei due player. Sono stati quindi aggiornati i campi 'winner_id' e 'loser_id' con i nuovi valori.
- 'Player_name': 19 nomi diversi di player presentavano 2 differenti id player. È stato quindi settato un unico id per ogni nome.
- 'Player_hand': 8 diversi player id presentavano 2 valori su 'hand'; ognuno di questi 8 presentava a volte un valore R (o L) e a volte il valore U di Undefined. Il valore U è stato sostituito con l'altro valore associato al player (R o L).
- 'Player_ioc': 4 diversi player id avevano 2 valori diversi su 'IOC'. Sono stati modificati guardando su wikipedia il valore corretto.
- 'Player_ht': 21 diversi player id presentavano 2 diversi valori su 'ht'. 20 di loro avevano valori diversi solo perchè mostravano valori null sull'attributo 'ht' su alcune righe; sono stati riempiti questi valori null con l'altezza trovata sulle altre righe. Un player presentava invece 2 valori diversi su 'ht' entrambi non null; per risolvere questa disuguaglianza è stato consultato wikipedia.
  I restanti missing value sull'altezza 'ht' sono stati riempiti con la media in base al sesso.

- 'Player_age': Age è un attributo che rappresenta l'età del player al momento del torneo, con l'età data dalla parte intera del valore e, nella parte decimale, la frazione (su 365) dei giorni che vanno dalla data di nascita alla data del torneo. Alcuni player presentavano il valore di age null su alcuni record ma non su altri; altri avevano invece valori null su tutti i record. In questo ultimo caso i valori null sono stati riempiti con la media attualizzata in base al sesso: cioè prima è stata calcolata l'età nell'anno 2021 di ogni giocatore diverso, poi è stata calcolata la media e successivamente si è calcolato il valore di age in base alla data del torneo. Il riempimento dei missing value non è stato eseguito per i player che presentavano almeno un valore non nullo; questo perché basta un valore di age per ogni player per calcolare l'anno di nascita necessario per il DW. Provando a calcolare l'anno di nascita a partire dai valori di Age e data del torneo, è risultato però che 58 player presentavano ognuno due valori diversi per l'anno di nascita. Ogni coppia differiva per 1 solo anno e, osservando bene i dati, si è giunti a dire che questo fatto è causato dalla non qualità di alcuni valori di Age, in particolare nella parte decimale del dato. Per rendere quindi il database corretto, è stato deciso di scegliere un anno di nascita per questi 58 giocatori e settare un valore Null su Age dove questo avrebbe portato al calcolo dell'altro anno di nascita. In questo modo ogni player, indifferentemente dall'ordine con cui è letto, avrà sempre lo stesso anno di nascita.
- 'Player_rank': in 'winner_rank' e 'loser_rank' i missing value sono stati riempiti utilizzando la media dello specifico player in questione se presentava altri valori su rank; oppure utilizzando la media delle medie dei vari players se il giocatore in questione non aveva mai nessun valore su rank.
- 'Player_rank_points': per 'winner_rank_points' e 'loser_rank_points' il riempimento dei valori mancanti è stato eseguito nello stesso modo degli attributi rank, ma utilizzando le medie su rank_points.
- 'Score': i missing value in 'score' sono stati riempiti con la moda.
- 'Tourney_level': per uno stesso torneo id sono associati spesso due livelli differenti del torneo. Questo analizzando i dati è probabilmente causato da tornei dove si giocano in contemporanea partite maschili e femminili, che presentano alcune volte una nomenclatura del livello differente. Per risolvere questo problema e lasciare il torneo id come chiave univoca, è stato deciso di sostituire l'id del torneo con la concatenazione dell'id del torneo + _ + livello del torneo.
- 'Tourney_id': 2 diversi tornei id presentavano ognuno 2 nomi diversi. Il primo è causato da un errore di digitalizzazione del dato, mentre l'altro presenta rappresenta l'id del torneo di qualificazione per gli Australian Open e cambia nome in base a femminile o maschile; si è deciso di lasciare solo la parte di nome comune.
- 'Torneo_surface': i missing value di surface sono stati riempiti con la moda.
- 'Torneo_match_number': erano presenti 21563 coppie di valori distinti di id torneo e match number con più di 2 righe uguali. Questo non avrebbe permesso di identificare la partita di uno specifico torneo in modo univoco. Sono stati quindi modificati tutti i valori di match_number incrementando ogni volta di 1 per lo stesso torneo, ripartendo da zero per ogni nuovo torneo.
- 'Match_statistics': molti attributi riferiti a statistiche di una singola partita (per esempio 'ace', 'df', 'svpt', '1stIn' ed altri) presentavano valori solo su 82284 record su 186073 totali ('minutes' 81636). Riempire così tanti missing value potrebbe portare ad una rappresentazione non veritiera della realtà e quindi è stato deciso di lasciare nulli questi valori.

1.1.2 Male_players.csv e Female_players.csv

- 'Player_name': 28 nomi di players presenti in tennis.csv non erano presenti né in male_players.csv né in female_players.csv. In aggiunta 64 nomi di player erano presenti sia in male_players che in female_players. Questi 64 nomi sono stati eliminati da entrambi file. Con la loro eliminazione il numero di players presenti in tennis.csv ma non in male_players.csv né in female_players.csv è salito a 34 (molti dei 64 nomi eliminati non si trovano in tennis). Questi 34 nomi sono stati inseriti in male_players o in female_players andando a vedere il sesso dell'avversario con cui disputavano la partita.

1.1.3 Countries.csv

Confrontando il csv fornitoci con gli ioc code distinti presenti in 'tennis.csv', si è notato che 30 codici su 154 non sono presenti in 'countries.csv'. Per recuperare le informazioni mancanti è stato utilizzato un csv esterno reperibile al link: https://github.com/johnashu/datacamp/blob/master/medals/Summer%20Olympic%20medalists%201896%20to%202008%20-%20IOC%20COUNTRY%20CODES.csv
Per i 30 ioc code mancanti sono state trovate le corrispondenze con i paesi che verranno poi utilizzate per recuperare il continente e la lingua rispettiva. 7 di questi ioc sono stati inseriti manualmente dalla lista presente su Wikipedia non essendo stati trovati nel csv esterno.

- 'Country_code' ITF: si tratta di un typo, infatti il player con ITF associato è un tennista italiano. Perciò è stato modificato in ITA in 'tennis.csv'
- 'Country_code' MRN: analizzando anche in questo caso le tenniste associate all'ioc si è scoperto che la nazione in questione è la Martinica (non presente nella lista di Wikipedia), perciò è stata inserita manualmente.

Per ottenere il continente e la lingua associata ad una nazione è stata utilizzata la libreria CountryInfo. Anche in questo caso i pochi record che la libreria non è stata in grado di riempire sono stati inseriti manualmente. Inoltre sono stati riscontrati i seguenti errori in 'countries.csv'

- 'Country_name' Urugay e New Zeland: corrispondono rispettivamente ad Uruguay e New Zealand, e sono stati modificati nel csv.
- 'Country_code' POC: che fa riferimento alle Filippine, conteneva il valore Unknown per country_name e continent.

Ottenuti i continenti e la lingua (è stata selezionata la prima lingua ufficiale) per tutti gli ioc, è stata droppata la colonna 'country_name' ed è stato esportata in csv la nuova versione di 'countries.csv'

1.2 Assignment 0

Con questo assignment sono state create le 5 tabelle necessarie per creare il DW su SQL Server Management Studio. Di seguito riportiamo i nomi delle tabelle con le chiavi primarie e le chiavi esterne al fine di mostrare lo schema generale. Per quanto riguarda i valori null, è stato deciso di renderli possibili su tutti gli attributi tranne che naturalmente per le chiavi primarie e le chiavi esterne.

- Date: Chiave primaria -> date_id
- Geography: Chiave primaria -> country_ioc
- Tournament: Chiave primaria -> tourney_id, Chiave esterna -> date_id (Date)
- Player: Chiave primaria -> player_id, Chiave esterna -> country_id (Geography)
- Match: Chiave primaria -> match_id, Chiavi esterne -> tourney_id (Tournament), winner_id (Player), loser_id (Player)

1.3 Assignment 1

In questo task l'obiettivo era quello di dividere i dati presenti nel file 'tennis.csv' in 4 file csv in accordo con lo schema del DW da realizzare. I 4 file da realizzare erano: 'match.csv', 'player.csv', 'tournament.csv' e 'date.csv'. Lo struttura del codice python utilizzato ('split.py') per questo compito è schematizzata di seguito; per cercare di renderlo efficiente ogni record di 'tennis_new' è stato letto una volta sola e sono state create delle strutture di supporto dati efficienti sia per recuperare il sesso di un player, utilizzando i dati presenti in 'male_new.csv' e in 'female_new.csv', sia per non andare a scrivere dati che si ripetono.

- Per leggere e scrivere i csv vengono utilizzate le funzioni della libreria csv DictReader e DictWriter. Con DictReader viene restituito un oggetto che si comporta come un reader normale ma che mappa

le informazioni di ogni riga su un dizionario, con il nome degli attributi come chiave. DictWriter permette invece di andare a scrivere nei file da creare ogni record direttamente come un dizionario.

- Creazione di due set, uno con i dati presenti in male e uno con quelli presenti in female. In entrambi i casi, i dati nome e cognome presenti su due colonne sono stati concatenati aggiungendo uno spazio tra di loro.
  Questi due set permettono di recuperare il sesso di un player andando a ricercare il nome in uno dei due set. La ricerca di un valore in set utilizza un algoritmo hashing e quindi avrà costo computazionale costante.
- Creazione di 3 dizionari inizialmente vuoti. Questi 3 dizionari saranno utili al fine di non inserire nei file da creare, dati duplicati. Ciò è possibile solo per i file tournament, player e date: i valori degli attributi presenti in questi file si ripetono molte volte in tennis e quindi servono delle strutture dati di supporto per gestire l'evenienza.
  Questi 3 dizionari conterranno ognuno a sua volta dei dizionari: ogni key del dizionario sarà infatti un valore della chiave primaria (univoca) per quel file, mentre il value sarà un dizionario con gli attributi del file come key e i valori degli attributi come value. Ogni volta quindi che sarà letto un nuovo record di tennis, basterà controllare se il valore della chiave primaria dei file da creare è già presente nel dizionario corrispondente. Questa ricerca costerà anche in questo caso tempo costante (utilizza un algoritmo hashing) e permetterà di capire se scrivere i dati del record oppure no.
- Scrittura delle righe di header nei 4 file da creare.
- Lettura una riga alla volta dei dati presenti in tennis. Per ognuno dei file da creare è richiamata una funzione diversa: queste funzioni (tranne che in match dove non ci sono problemi di record duplicati), vanno a vedere grazie ai dizionari di supporto, se i dati presenti sulla la riga attuale, sono già stati scritti; nel caso esegue soltanto dei controlli per verificare che per record con stessa chiave primaria non ci siano differenze negli altri valori. Se invece la chiave non è presente nel dizionario allora vengono recuperati i dati da scrivere nel file csv specifico; essi vengono inseriti in un dizionario in modo sia da scriverli direttamente nel file csv (DictWriter fa scrivere un record partendo da un dizionario) sia da inserirli nel dizionario di supporto.
- Due valori da inserire in player non sono presenti come attributi in tennis: sex e year_of_birth. Per recuperare i valori di sex sono stati utilizzati i due set di supporto creati da male e female. In questo modo è bastato ricercare in quale dei due set fosse presente il nome del player.
  Per recuperare year_of_birth l'operazione è stata più complessa ed è perciò stato necessario creare una funzione apposita: in tennis è presente il valore age, che si riferisce all'età del player durante il torneo che sta giocando. Inoltre, in age abbiamo un valore float composto dalla parte intera che corrisponde al numero di anni del player e la parte decimale che corrisponde alla frazione (su 365) dei giorni che vanno dalla data di nascita alla data del torneo. Per calcolare l'anno di nascita è stato necessario anche tenere in considerazione questi giorni per evitare problemi con date di nascita e date del torneo a cavallo tra due anni diversi (es. Dicembre e Gennaio).
- Close di tutti i file aperti inizialmente.

1.4 Assignment 2

L'ultimo assignment consisteva nell'inserire i record presenti nei csv, splittati a partire da 'tennis.csv', sul database (codice python 'insert.py').
Per far ciò abbiamo definito una funzione insert che prende come parametri: la connessione, il reader e il nome della tabella del database su cui effettuare l'insert. La funzione in questione effettua prima una query per ottenere le colonne di quella specifica tabella dalla vista INFORMATION_SCHEMA.COLUMNS e successivamente effettua l'insert con una query parametrica.

1.5 Contenuto della cartella

La cartella compressa conterrà al suo interno:

- 'preprocessing.ipynb', contenente il notebook nel quale viene effettuato il preprocessing. Quest'ultimo utilizzerà 'tennis.csv', 'male_players.csv', 'female_player.csv' e 'countries.csv' presenti nella cartella 'data' (lasciata vuota nello zip per non appesantire il file, andranno quindi caricati i csv menzionati). E produrrà 'tennis_new', 'male_new' e 'female_new' nella cartella 'data' e 'countries_new' nella cartella 'table' (già pronta per essere inserita nel database).
- 'split.py', contenente il codice per splittare 'tennis_new.csv' in 'match.csv', 'tournament.csv', 'date.csv' e 'player.csv' in 'table'
- 'insert.py', che contiene il codice per caricare le tabelle della cartella 'table' sul database

Parte 3

2.1 Creazione Cubo

Struttura cubo

Dopo aver popolato il DW si è proceduto con la creazione del Cubo. Innanzitutto è stato necessario creare le seguenti dimensioni:

- Match
- Tournament
- Player
- Geography
- Date

Nell'implementazione le dimensioni Date e Geography sono state inserite successivamente al wizard di creazione del cubo, in questo modo è possibile tenerle separate dalle dimensioni Tournament e Player rispettivamente. Dal tab 'Utilizzo dimensioni' sono stati settati i tipi di relazioni delle dimensioni.

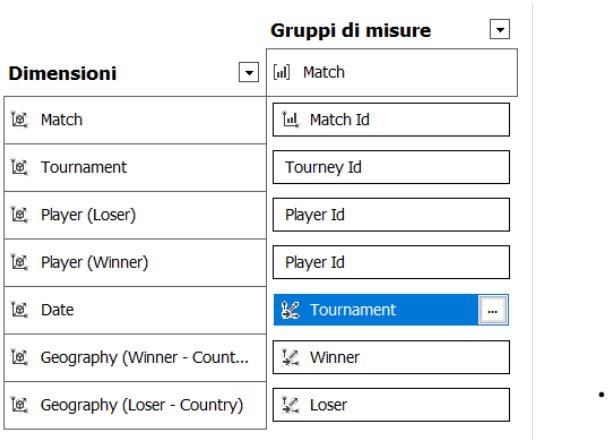

Figure 2.1: Utilizzo dimensioni

Per quanto riguarda Player vengono create 2 dimensioni derivate: Winner per la surrogate key winner_id e Loser per la surrogate key loser_id.
Match è stata definita di tipo 'Fatto' essendo la fact table, mentre Date e Geography sono state definite con relazioni di tipo 'Riferimento' di Tournament e Player rispettivamente. Anche in questo caso vengono create 2 dimensioni derivate da Geography per Winner e Loser.
Le measure utilizzate sono le seguenti:

- Winner Rank
- Winner Rank Points
- Loser Rank
- Loser Rank Points
- Conteggio di Match
- Conteggio di valori univoci di Winner Id
- Conteggio di valori univoci di Loser Id

Il conteggio di Player distinti sarà poi utile in seguito nella creazione delle Dashboard dove sono risultati utili anche i seguenti membri calcolati:

- Average Loser Rank ([Measures].[Loser Rank]/[Measures].[Conteggio di Match])
- Average Loser Rank Points ([Measures].[Loser Rank Points]/[Measures].[Conteggio di Match])
- Average Winner Rank ([Measures].[Winner Rank]/[Measures].[Conteggio di Match])
- Average Winner Rank Points ([Measures].[Winner Rank Points]/[Measures].[Conteggio di Match])

Struttura dimensioni

La fact table Match contiene alcuni attributi che contengono valori Null perciò è stato necessario settare la proprietà UnknownMember a Visible e per tutti gli attributi contenenti Null settare NullProcessing a UnknownMember. A questo punto nel tab 'Browser' è possibile notare come siano presenti anche valori con dicitura Unknown.

Per quanto riguarda Date sono state create 2 colonne supplementari nella vista per ordinare mesi e giorni (order_month e order_day). Come specificato nel testo dell'Assignment è stata creata la gerarchia Time mostrata in figura 2.2. Stessa cosa per Geography con la gerarchia ContinentCountry (figura 2.3)

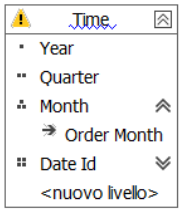

Figure 2.2: Gerarchia Time

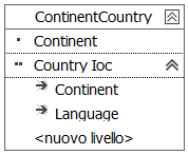

Figure 2.3: Gerarchia ContinentCountry

Niente da aggiungere per Player e Tournament che invece non presentavano gerarchie.

2.2 MDX

Assignment 1

"Show the player that lost the most matches for each country"

L'obbiettivo di questa query era quello di disporre su ogni riga una tupla formata da una nazione e il nome del player di quella nazione con il numero maggiore di match persi (considerando la specifica nazione). Non dovevano essere presenti nazioni che presentavano player con sole vittorie e conseguente valore null sul numero di match persi (condizione difficile da verificarsi ma non impossibile).

Assignment 2

"For each tournament, show the loser with the lowest total loser rank point"

Il risultato voluto da questa query era quello di disporre su ogni riga una tupla formata dall' id del torneo e il nome del player con il più basso valore su total loser rank (considerando lo specifico torneo).

Assignment 3

"For each tournament, show the loser with the highest ratio between his loser rank point and the average winner rank points of that tournament"

Per questa query l'obbiettivo era quello di presentare per ogni riga una tupla con l'id del torneo e il nome del player con il più alto rapporto tra il suo loser rank point e la media del punteggio di winner rank point sul torneo considerato.

2.3 Dashboard

Assignment 4

"Create a dashboard that shows the geographical distribution of winner rank points and loser rank points"

Per rappresentare la distribuzione delle misure winner rank point e loser rank point sono state realizzate 2 dashboard separate ma con la stessa struttura (file 'Assignment4'). Nella pagina 1 del file è presente la dashboard per la misura winner rank points, mentre nella pagina 2 è presente quella per loser rank points. Sono stati utilizzati 4 grafici: 'mappa', 'istogramma a colonne in pila', 'grafico a torta' e 'scheda'. Scheda è utilizzato solo per visualizzare il valore di winner rank points (o loser rank points). Negli altri 3 grafici è invece presente la gerarchia costituita da continente e country IOC, che permette di effettuare analisi interattive. E' inoltre possibile fissare un valore su un grafico e ottenere un filtro per quel valore sugli altri 2. Può risultare particolarmente utile questa funzione, ad esempio per fissare un certo continente su mappa ed analizzare poi sul grafico a torta (o sull'istogramma) i valori dei country di tale continente.

Assignment 5

"Create a plot/dashboard of your choosing, that you deem interesting w.r.t the data available in your cube"

Per completare questo task è stato deciso di realizzare 2 dashboard diverse, riportate rispettivamente nei file 'Assignment5_pagina1' e 'Assignment5_pagina2'.

La prima ('Assignment5_pagina1') permette di visualizzare la media della misura Winner Rank in base all'anno di nascita. Questo permette di effettuare un'analisi al fine di osservare come cambia la posizione in classifica dei vincitori in base al loro anno di nascita. Sono presenti anche due istogrammi che riportano il numero di distinct winner player in base al sesso ed in base alla hand. Questi istogrammi possono essere utili oltre che per vedere le differenze nel numero di giocatori in base al sesso ed in base alla mano, anche per vedere il numero di player diversi nati in un certo anno a cui si riferisce il valore di Average Winner Rank. Sono presenti anche un grafico mappa per effettuare analisi anche in base a continente o nazione e dei filtri da poter inserire su sex, hand e date.

La seconda dashboard ('Assignment5_pagina2') permette invece di andare a selezionare un player da una lista per visualizzarne le sue statistiche. In particolare, il suo valore medio di Winner Rank, cioè alla sua posizione in classifica ed a come essa varia negli anni. Sono presenti anche svariati filtri al fine di rendere l'utente libero di effettuare la ricerca che ritenga più opportuna.