Analisi delle performance studentesche

a cura di

S. Caruso, C. Monaco, S. Monaco

Data Science and Machine Learning

Prof. Giuseppe Polese

Università degli Studi di Salerno

a.a. 2018/19

Abstract

Viviamo in un mondo interconnesso, dove ogni giorno vengono prodotte quantità esagerate di dati. L'enorme quantità di dati che l'essere umano deve immagazzinare ha rappresentato anziché un problema un'opportunità. Infatti, analizzando e studiando i dati è possibile estrarre conoscenza da essi al fine di migliorare qualche aspetto di vita quotidiano. L'obiettivo di questa tesi è proprio quello di studiare e condurre esperimenti sui dati relativi alle carriere degli studenti del Dipartimento di Ingegneria dell'Università degli Studi di Salerno, al fine di estrapolare quella conoscenza adatta per migliorare l'offerta didattica del dipartimento e per fornire supporto agli studenti nelle fasi più critiche della propria carriera.

Nel lavoro di tesi che segue, verranno presi in considerazione due momenti critici della carriera di uno studente, l'approccio ad un nuovo corso di studio e la capacità del singolo di concludere lo stesso nei tempi stabiliti. L'obiettivo sarà, tramite l'utilizzo delle moderne tecniche di Machine Learning, comprendere se esiste o meno una correlazione tra il background di uno studente (in termini di tipologia di formazione liceale e votazione riportata) e le performance ottenute dallo stesso durante la sua carriera universitaria.

Sommario

Capitolo 1

Introduzione

Analizzare i dati relativi alle carriere degli studenti presenta un'ottima opportunità per tutti gli stakeholders coinvolti. Dal punto di vista del dipartimento di riferimento, analizzare l'andamento delle carriere dei propri studenti permetterebbe di comprendere se esistono problemi e difficoltà in cui gli studenti incorrono ed in che modo essi possono impattare sul proseguimento degli studi. Intervenire strategicamente nei punti critici, permetterebbe al dipartimento di innalzare la qualità media dell'offerta formativa e di conseguenza innalzare il livello di preparazione che esso può offrire. Ad esempio, se dall'analisi dei dati, si evincerebbe che una buona percentuale di studenti con un determinato background liceale ha problematiche nel conseguimento di un cospicuo numero di esami al primo anno, permetterebbe al dipartimento una ristrutturazione del piano didattico al fine di supportare e non "lasciare indietro" i suddetti studenti.

Per gli studenti, invece, un'analisi del genere rappresenterebbe una buona opportunità di far sentire la propria. È assai diffuso, infatti, un comportamento riservato dello studente che sta affrontando delle difficoltà nel suo percorso universitario. Per quanto in un primo momento potrebbe portare l'individuo a sentirsi banalmente un numero, come se fosse una macchina, sapere che la difficoltà che sta incontrando è diffusa tra gli studenti simili a lui e sapere che il suo dipartimento si sta muovendo per arginarla rappresenterebbe una speranza ed un incentivo ad andare avanti con più determinazione.

Inoltre, è da non sottovalutare anche la sensazione percepita da tutti gli studenti in generale, che il proprio dipartimento è attento alle proprie necessità e cerca di migliorare l'esperienza di studio in generale.

Gli obiettivi che sono stati fissati dal Dipartimento di Ingegneria, nella persona del Prof. Ing. Stefano Riemma, rappresentano un ottimo punto di partenza per questo percorso di analisi e monitoraggio che ci auguriamo perduri nel tempo. Infatti, l'Ing. ci ha chiesto di analizzare i dati e comprendere dapprima, se ci fosse una correlazione tra il background dello studente e il numero di esami che lo studente sarà in grado di sostenere al primo anno.

Appurata l'esistenza o meno di tale correlazione, l'altro obiettivo che è stato fissato è quello di comprendere se in relazione alla formazione dello studente ed al numero di esami sostenuti al primo anno si possa già determinare se lo studente in questione ha buone probabilità di terminare il proprio percorso di studio entro i termini stabiliti dall'ordinamento di riferimento o meno.

CFU – credito formativo universitario

In Italia, il sistema accademico organizza la distribuzione delle attività didattiche basandosi su un parametro univoco -a livello nazionale- che rappresenti l'effettivo carico di lavoro dell'attività presa in esame. Tale parametro prende il nome di CFU, ovvero Credito Formativo Universitario. Ogni CFU corrisponde ad un carico di lavoro pari a 25 ore. Ogni Università, e più nello specifico, ogni dipartimento, assegna ad ogni attività didattica un numero variabile di CFU, in base all'impegno ritenuto necessario al fine del superamento dell'attività didattica stessa. In media le attività didattiche universitarie sono suddivise in tre categorie -distinte dal numero di CFU necessari- le quali sono:

- 6 (sei) CFU
- 9 (nove) CFU

- 12 (dodici) CFU

È chiaro che possono essere presenti altri casi (come ad esempio attività da 3 CFU o da 15).

Nello specifico, le università italiane strutturano il corso basandosi sul carico didattico attribuito dal numero di CFU. Infatti, le ore di lezione che il dipartimento eroga allo studente per una determinata attività didattica varia a seconda proprio di quanto l'attività è percepita impegnativa per lo studente, da parte del dipartimento stesso. Infatti, le 25 ore del singolo CFU vengono così suddivise:

- 30% attribuito alle attività teoriche
- 50% attribuito alle attività teoriche-pratiche

100% nel caso di attività di laboratorio

La normativa nazionale non stabilisce alcun vincolo di organizzazione, ma lascia spazio all'Ateneo di strutturare al meglio le proprie attività, impostando comunque un tetto massimo di incremento o diminuzione del monte ore attribuito al singolo CFU del 20%.

Sempre analizzando la normativa nazionale, si evince che la quantità media di impegno di apprendimento, svolto in un anno da uno studente a tempo pieno, è convenzionalmente fissata in 60 crediti (CFU).

In conclusione, è bene specificare che per il completamento di un percorso di studi di Laurea Triennale, in Italia è necessario il raggiungimento di 180 (centottanta) CFU, per una Laurea Magistrale è necessario il raggiungimento di 120 (centoventi) CFU, mentre per un Diploma di Laurea Magistrale è necessario il raggiungimento di 300 (trecento) CFU.

In corso o fuori corso

Il percorso di studi accademico in Italia viene pianificato distribuendo le attività didattiche necessarie al completamento in più anni di corso. Esistono principalmente tre tipologie di corso di studi:

- Laurea Triennale – della durata di appunto 3 anni
- Laurea Magistrale – della durati di 2 anni, ci si accede solo dopo il conseguimento di una Laurea Triennale attinente

Diploma di Laurea Magistrale o Laura Magistrale a ciclo unico – della durata di 5 anni

Ne consegue che lo studente deve completare tutte le attività didattiche previste dal suo corso di studio nei tempi stabiliti. Se ciò non accadesse, e quindi al termine della durata prevista allo studente manca il superamento di alcune attività didattiche, si dice che si è in fuori corso. Essere in fuori corso sostanzialmente vuol dire che lo studente non è stato in grado di completare anno per anno le attività didattiche proposte dal Dipartimento e che quindi ha necessità di recuperarle impiegando più tempo di quanto stabilito.

Andare fuori corso comporta delle conseguenze (più o meno gravi, dipende dal punto di vista) sia per lo studente che per il Dipartimento.

Per quanto riguarda lo studente, le conseguenze più note sono l'aumento delle tasse universitarie e la perdita dei punti bonus che si possono percepire all'esame finale.

Per il Dipartimento invece, avere un numero di studenti in fuori corso comporta una perdita di punti a livello nazionale, e di conseguenza la discesa in classifica. Essere considerati un buon dipartimento a livello nazionale non comporta

solamente orgoglio e gloria, ma bensì un maggiore approdo di fondi da destinare alla ricerca e al potenziamento delle attività didattiche.

Ne discende quindi che entrambe le parti sono interessate ad evitare il fuori corso.

Capitolo 2

Lo stato dell'arte

Per quanto innovativa sembri questa iniziativa non è la prima volta che l'utilizzo di tecniche di Machine Learning viene destinato all'analisi di performance delle carriere universitarie.

Nel Survey di Shahiri, Husainm e Rashid [1] è presente una raccolta di numerosi tentavi di effettuare predizioni sulle performance degli studenti utilizzando le tecniche di Data Mining. Nell'articolo è possibile leggere di diversi approcci ognuno dei quali si basa su informazioni differenti.

Predire l'andamento della carriera universitaria di uno studente non è un task affatto semplice. Entrano in gioco infatti, numerosi fattori, che vanno dal livello di preparazione del singolo fino ad esplorare il contesto sociale nel quale esso vive.

È interessante comprendere come a seconda delle informazioni che si posseggono si sceglie la strategia da adottare ed in particolare quali tecniche di Machine Learning risultano essere efficienti.

I fattori più importanti per la predizione dell'andamento universitario di un individuo sono:

- CGPA – Cumulative Grade Point Average
- Demografica dello Studente
- Informazioni extra-scolastiche
- Psicometria dell'individuo

CGPA – cumulative grade point average

Il CGPA è un sistema molto utilizzato all'estero per calcolare un punteggio sull'andamento scolastico dello studente. Sostanzialmente è la media della votazione riportata in tutte le attività didattiche sostenute dallo studente.

Facciamo un esempio di come si calcola il CGPA:

Supponiamo che uno studente frequenti 5 attività didattiche: Matematica, Fisica, Chimica, Disegno e Informatica e riporti le seguenti valutazioni:

- Matematica = 27
- Fisica = 28
- Chimica = 25
- Disegno = 30
- Informatica = 30

Il totale è 27 + 28 + 25 +30 + 30 = 140. Per calcolare il CGPA 140/5 = 28.

Nel sistema italiano il CGPA rappresenta la media aritmetica dei voti riportati.

Nei lavori esaminati il CGPA [2, 3, 4, 5, 6, 7, 8, 9, 10, 11] è di gran lunga l'informazione più utilizzata per effettuare una predizione sulle performance di uno studente. Questo poiché è un dato tangibile di come lo studente sta approcciando allo studio e come sta reagendo in relazione alle nuove nozioni che sta apprendendo.

Diventa chiaro che quanto più il CGPA è alto più è facile capire che lo studente non sta affrontando difficoltà e potrà con serenità terminare gli studi nel periodo stabilito.

Viceversa, se il CGPA di uno studente è relativamente basso, oppure se subisce una discesa da un anno all'altro, allora lo studente sta incontrando qualche difficoltà nel percorso accademico oppure nella vita extra-scolastica.

In conclusione, il CGPA può essere definito come il dato più rappresentativo del percorso di uno studente.

Demografica dello studente

Molti lavori sull'argomento prendono in considerazione la demografica dello studente [2, 7, 12, 13, 14, 15].

È stato notato infatti che il sesso dello studente, in linea generale, contribuisce molto all'andamento scolastico. Nel lavoro di S. Meit, N.J. Borges, B. Cubic e H. Seibel, [20] viene evidenziato come "le studentesse hanno più senso del dovere, sono più sensibili, più autosufficienti, più concentrate verso l'apprendimento, più organizzate e autodisciplinate nei confronti degli studenti. Gli studenti, invece sono più propensi all'adattamento, più determinati, più sospettosi e scettici, più fantasiosi e orientati all'intuizione, più discreti e individualisti e più rispettosi verso altri studenti, rispetto alle studentesse."

È evidente come la differenza di sesso incide a più livelli nel momento dell'affronto di una difficoltà. Ad esempio, uno studente riuscirà a dare il meglio anche in contesti che non lo soddisfano al 100%, mentre le studentesse riusciranno a dare il meglio anche in situazioni dove percepiscono poco supporto da parte degli insegnanti.

Situazioni del genere, sono tutt'altro che rare e un modo di reagire così diverso incide sicuramente sul percorso universitario.

Di conseguenza conoscere il sesso e informazioni generiche sulla personalità dello studente rappresenta un'ottima informazione se si vuole predire l'andamento accademico.

Informazioni extra-scolastiche

Un altro parametro importante per questo task di predizione è la raccolta di informazioni che non riguardino l'andamento scolastico dell'individuo, come ad esempio la distanza dall'università, il reddito familiare e più in generale il contesto in cui vive.

È bene ricordare infatti che oltre alla vita accademica lo studente è sottoposto a innumerevoli altre sollecitazioni esterne che possono influire su quello che poi sarà l'andamento scolastico. Ad esempio, se uno studente trascorre troppo tempo in viaggio per arrivare in sede, si sentirà stanco e demotivato nell'affrontare lo studio, oppure ad esempio se lo studente ha necessità di lavorare per completare gli studi si sentirà allo stesso modo stanco e stressato e finirà per dedicare meno tempo allo studio di quanto vorrebbe. Seppur sembrino ovvietà, intervistando un considerevole numero di studenti, ci siamo subito resi conto che queste informazioni potrebbero essere state molto utili al fine di portare al termine il nostro task.

Psicometria dell'individuo

Forse il più scontato dei fattori, ma in psicologia è noto che gli interessi della persona influiscono molto sull'atteggiamento che ha nei confronti delle cose che fa.

Non è difficile che uno studente si renda consapevole troppo tardi che il percorso di studi che ha scelto non è la cosa che più lo interessi.

Sapere gli interessi, le attitudini, le passioni, e più in generale l'orientamento della persona verso lo studio di determinate discipline, possono risultare dei fattori importanti.

Diventa subito evidente infatti che se uno studente non interessato agli argomenti che sta apprendendo, ottiene brutti risultati, o inceppa in qualche difficoltà, ne uscirà demotivato e non riuscirà ad affrontare con la giusta determinazione il percorso di studi, influenzando quelle che poi saranno le sue performance.

Essere a conoscenza delle attitudini del singolo studente, in molti studi [16, 17, 18, 19] ha rappresentato un vantaggio nel determinare con quali probabilità lo studente porta a termini il percorso che ha scelto oppure rinuncia.

Capitolo 3

Il dataset

Il dipartimento di Ingegneria, nella persona del Prof. Ing. S. Riemma, ci ha fornito un dataset riguardante le carriere degli studenti iscritti alla propria facoltà. Una buona parte del dataset è composta dagli studenti del primo ciclo (Laurea Triennale) ed un'altra parte dagli studenti del secondo ciclo (Laurea Magistrale).

Di seguito verranno elencati tutti gli attributi (le informazioni) presenti per ogni studente.

- Matricola: Rappresenta la matricola dello studente, ovvero l'identificativo univoco che il dipartimento associa allo studente. Nel rispetto della normativa privacy, questo dato è stato sostituito con un intero progressivo.

- Codice Fiscale: Il codice fiscale della persona. Allo stesso modo della Matricola anche quest'informazione è stata modificata per il rispetto della privacy. Il contenuto di questo campo è una stringa alfanumerica.

- 1: Questo campo rappresenta il numero di esami sostenuti dallo studente nel primo anno di corsi. L'informazione viene espressa sotto numero di CFU sostenuti

- 2: Questo campo rappresenta il numero di esami sostenuti dallo studente nel secondo anno di corsi. L'informazione viene espressa sotto numero di CFU sostenuti

- 3: Questo campo rappresenta il numero di esami sostenuti dallo studente nel terzo anno di corsi. L'informazione viene espressa sotto numero di CFU sostenuti

- Tot: Questo campo rappresenta il numero di esami sostenuti dallo studente alla scadenza del suo percorso regolare. L'informazione viene espressa sotto numero di CFU sostenuti

- CDS: Stringa numerica che esprime la tipologia del Corso Di Studio che lo studente frequenta.

- Tipo CDS: Rappresenta a quale ciclo di studio lo studente è iscritto, ovvero, se triennale oppure magistrale.

- Coorte: Indica l'anno in cui lo studente si è immatricolato al suo corso di studio.

- Anni Carriera: Esprima quanto è durata la carriera universitaria dello studente.

- Anno_Diploma: Indica l'anno in cui lo studente ha conseguito il diploma di scuola superiore

- Voto_Diploma: Indica il voto riportato dallo studente al suo diploma di scuola superiore

- Codice_Meccanografico: Codice univoco che rappresenta la scuola dove lo studente ha conseguito il suo diploma

- Tipo_Maturità: La tipologia di diploma che lo studente ha perseguito

- Anno_Accademico_Laurea: L'anno accademico in cui lo studente ha conseguito il diploma di laurea

- Voto_Laurea: Il voto conseguito al diploma di laurea

- Erasmus: Variabile booleana che esprime se lo studente ha aderito o meno ad un'iniziativa di mobilità all'estero.

- Tesi_Estero: Variabile booleana che esprime se lo studente ha sostenuto o meno il percorso di tesi all'estero

- Stato_Studente: Attributo che esprime lo stato attuale della carriera dello studente.

- Motivo_Stato_Studente: Indica il motivo per il quale la carriera dello studente si trova in un determinato stato.

Dall'analisi di questi attributi è subito evidente di come il dataset non sia in possesso delle informazioni più importanti prese in considerazione al Capitolo 2.

Manca infatti, qualsiasi informazione inerente al calcolo del CGPA, che come visto è di gran lunga l'informazione necessaria se si vuole fare predizione delle performance universitarie.

L' unica informazione utile a determinare il livello di preparazione dello studente al momento dell'iscrizione al corso di studio è rappresentata dal voto del diploma.

Ne consegue che affidandosi a quest'unica informazione potrebbe essere molto facile andare in underfitting, ovvero un qualsiasi modello non sarebbe in grado di comprendere la realtà (il livello di preparazione dello studente), al fine di predire quanti CFU verranno sostenuti dallo studente.

Inoltre, le informazioni di cui il dataset dispone ignorano completamente il fattore persona.

Non vi è alcuna informazione utile, infatti, per comprendere che tipo di persona è lo studente. Non è possibile capirne gli interessi, le motivazioni, le condizioni sociali.

In aggiunta, la distribuzione dei valori per alcuni attributi non è uniforme e non si avvicina nemmeno ad una distribuzione equa.

Infatti, oltre il 73% degli studenti presenti ha conseguito un diploma di Liceo Scientifico. Il 63% del dataset rappresenta studenti che si sono laureati nel tempo previsto, e quindi non sono andati "Fuori Corso". Circa il 36% degli studenti si è diplomato con un voto compreso tra il 95 e il 100 (e se estendiamo alla fascia 90-100, il dato cresce al 48%).

Tutto questo renderà ovviamente complicato ottenere buoni risultati in fase di predizione.

Preso atto delle problematiche relative alla qualità delle informazioni che il dataset offre, doverosamente bisogna considerarne anche la quantità.

Il dataset che ci è stato fornito, pur essendo ampio "orizzontalmente" (numero di attributi) pecca per quello che riguarda l'ampiezza "verticale".

Infatti, in tutto il dataset dispone di solamente 5652 tuple, che come già detto prima sono spalmate su due cicli di studi. Ovviamente non ha un significato dal punto di vista semantico mischiare le due tipologie di carriere per portare a termine i nostri task.

Questo perché gli studenti che si accingono a completare il primo anno del secondo ciclo (magistrale), hanno un background totalmente diverso da quelli che provengono dal liceo e stanno per iniziare il percorso universitario.

Inoltre, il dataset non è privo di valori nulli e poco significativi, che andranno eliminati.

Va considerato che in alcuni casi la pulizia del dataset, ovvero la fase in cui vengono corretti eventuali inconsistenze dei dati, non sarà possibile, vedi ad esempio le tuple per cui mancano il voto del diploma, o il numero di CFU sostenuti al primo anno.

Esposte tutte queste problematiche al committente del progetto, ci è stato consigliato di continuare comunque nel lavoro per fornire quantomeno una base per eventuali lavori futuri, e cercare comunque di comprendere se ci fossero anche degli accenni di correlazione.

Data preparation

Prima di iniziare l'analisi dei dati con le tecniche di Machine Learning, si è resa necessaria una fase di pulizia e preparazione dei dati.

La prima problematica che abbiamo affrontata è stata quella di rendere coerenti le informazioni che avevamo a disposizione. Abbiamo dovuto quindi eliminare

dal dataset tutte le tuple che presentavano valori nulli nei campi d'interesse. Molte tuple infatti mancavano di valori nel campo "1", ovvero nell' attributo target del nostro primo task. Non avendo modo di riuscire a quantificare l'attributo abbiamo dovuto per forza non utilizzare quella tupla nelle nostre analisi.

Successivamente abbiamo notato la presenza di molte varianti di tipologie di diploma. Infatti, ad esempio, vi erano molte varianti del diploma di istituto tecnico. Ai fini della nostra analisi, sapere quale indirizzo specifico è stato scelto dallo studente al diploma poteva risultare superfluo ed inoltre avrebbe diminuito il numero di istanze di studenti dello stesso diploma (in questo caso Istituto Tecnico) e di conseguenza abbassato il grado di precisione. Essendo il livello di preparazione, ai fini universitari, molto simile per quanto riguarda lo stesso diploma, abbiamo eliminato tutte le varianti di un diploma, sostituendole con il Diploma "madre" concernente. Questo ci ha permesso di passare da 62 varianti ai 13 diplomi principali, aumentando così il numero di istanze per ogni diploma.

Tutti i modelli che il nostro team è stato incaricato di sviluppare prediligono il lavoro su dati in formato numerico. La maggior parte delle feature del nostro dataset erano in formato alfanumerico o prettamente alfabetico (stringhe di caratteri). Prima di cominciare a lavorare di conseguenza abbiamo dovuto mappare i diversi valori alfabetici-alfanumerici in valori numerici. Per farlo sono state utilizzate prettamente due tecniche:

1. Funzione Hash del valore – in questo modo il valore veniva sostituito con un numero intero di 8 cifre. Per mantenere il valore costante durante tutte le computazioni, veniva calcolato l'hash una singola volta e scritto il risultato in un file, che negli utilizzi futuri veniva letto. Questa tecnica è stata successivamente scartata, poiché avere un intero di 8 cifre che

rappresentava informazioni utili in fase di addestramento poteva fuorviare il modello, soprattutto considerando gli algoritmi che forniscono in output una funzione lineare (i modelli di regressione).

2. Enumerazione del dizionario – tutti i valori di un determinato campo sono stati inseriti in un dizionario, successivamente abbiamo fatto un'enumerazione di tutti i valori. In questo modo per ogni attributo vi era una mappatura univoca in valori compresi tra 1 e il dominio del campo stesso (ovvero se il campo avesse presentato 100 valori possibili, le enumerazioni sarebbero state comprese tra 1 e 100).

Nella fase di sviluppo di alcuni modelli, ci sono state delle piccole modifiche per adattare i dati al modello stesso. Discuteremo delle modifiche fatte nel seguente capitolo, nella sezione relativa al rispettivo modello.

Capitolo 4

Il machine learning

Quando si parla di machine learning si parla di una particolare branca dell'informatica che può essere considerata una parente stretta dell'intelligenza artificiale e si pone l'obiettivo di far apprendere in modo automatico attività alle macchine, svolte da noi esseri umani. È un ramo molto vasto e prevede quindi differenti modalità, tecniche e strumenti. L'apprendimento automatico raccoglie un insieme di metodi, sviluppati a partire dagli ultimi decenni in varie comunità scientifiche, sotto diversi nomi: statistica computazionale, riconoscimento di pattern, reti neurali artificiali, filtraggio adattivo, teoria dei sistemi dinamici, elaborazione delle immagini, data mining, algoritmi adattivi, ecc.. quindi utilizza metodi statistici per migliorare progressivamente la performance di un algoritmo nell'identificare pattern nei dati.

Il termine Machine Learning fu coniato nel 1959 da Arthur Samuel e ripreso successivamente da Tom Mitchell che ne ha dato una definizione formale:

"Si dice che un programma impara da una certa esperienza E rispetto a una classe di compiti T ottenendo una performance P, se la sua performance nel realizzare i compiti T, misurata dalla performance P, migliora con l'esperienza E."

Significa che un computer impara nel momento in cui migliora nello svolgimento di un task rispetto alla sua esperienza. Tramite l'apprendimento creiamo delle regole generali che sono assimilabili in modelli di apprendimento. L'apprendimento è un processo iterativo, che continua per tutta la vita e ci permette di migliorare le nostre conoscenze a seconda delle informazioni che raccogliamo. Lo stesso fanno le macchine: dai dati in input (generalmente si parla

di Big data) si ricavano i modelli di apprendimento. Quest'ultimi permettono di costruire algoritmi di apprendimento per risolvere uno specifico problema.

L'algoritmo indica alla macchina le operazioni che può eseguire e che cosa può fare, come ad esempio, riconoscere un cane da un'immagine. Quando è in grado di farlo, utilizzerà tale informazione per analizzare le successive immagini.

I sistemi di apprendimento automatico possono essere classificati in base alla quantità e al tipo di supervisione che ottengono durante l'allenamento. Esistono quattro categorie principali:

- Apprendimento supervisionato: è una tecnica di apprendimento automatico che mira a istruire un sistema informatico in modo da consentirgli di elaborare automaticamente previsioni sui valori di uscita di un sistema rispetto ad un input sulla base di una serie di esempi ideali, costituiti da coppie di input e di output, che gli vengono inizialmente forniti.

- Apprendimento non supervisionato: è una tecnica di apprendimento automatico che consiste nel fornire al sistema informatico una serie di input (esperienza del sistema) che egli riclassificherà ed organizzerà sulla base di caratteristiche comuni per cercare di effettuare ragionamenti e previsioni sugli input successivi. Al contrario del supervisionato, durante l'apprendimento non vengono forniti esempi, questo perché non sono noti e l'obiettivo è proprio quello di individuare classi automaticamente.

- Apprendimento semi supervisionato: non è altro che il connubio delle due tecniche pocanzi elencate.

- Apprendimento di rinforzo: è una tecnica di apprendimento automatico che punta ad attuare sistemi in grado di apprendere e adattarsi alle mutazioni dell'ambiente in cui sono emersi, attraverso la distribuzione di una "ricompensa" detta rinforzo che consiste nella valutazione delle loro prestazioni.

I nostri obiettivi

Per il raggiungimento degli obiettivi stabiliti, il nostro team ha utilizzato modelli di apprendimento supervisionati e non supervisionati, nello specifico si è occupato di utilizzare modelli di regressione e clusterizzazione.

Nei seguenti paragrafi saranno presentati tutti i modelli protagonisti del nostro studio e per ognuno verrà esposta una breve introduzione dello stesso e come sono stati utilizzati per i due task, che ricordiamo:

- Task1: individuare l'esistenza di correlazione (o meno) tra il background dello studente e l'inizio del percorso di studi dello stesso. Nello specifico si vuole rispondere alla domanda, date le features "Tipo diploma", "Voto di maturità" è possibile predire il numero di cfu che lo studente conseguirà il primo anno?

- Task2: individuare l'esistenza di correlazione tra il background dello studente, l'inizio del percorso di studi e la conclusione del percorso di studi. Nello specifico date le feature precedenti, "Tipo diploma"," Voto maturità" e con l'aggiunta dei "cfu conseguiti al primo anno" posso predire se lo studente concluderà i suoi studi in corso?

Regressione

La regressione è una tecnica che viene usata per analizzare una serie di dati, con l'obiettivo di trovare una forma di relazione funzionale tra di essi. Più nello specifico, i problemi di regressione sono caratterizzati da due tipi di variabili:

1. Variabili Dipendenti: Le variabili (attributi) per il quale si vuole individuare la relazione funzionale

2. Variabili Indipendenti: Le variabili (attributi) che fungono da "input" per la relazione funzionale

Informalmente, la variabile indipendente è detta "attributo target", dato che l'obiettivo è proprio quello di stimarne il valore tramite una relazione funzionale con le variabili indipendenti, che a loro volta vengono chiamate appunto, "attributi predittivi". È importante ricordare che nella costruzione di questo modello è da considerare che essi sono soggetti ad un termine di errore (noise) che è una variabile casuale rappresentante proprio un possibile errore nel calcolo della relazione.

Gli errori possono avere molteplici fonti, ad esempio un rumore nei dati (se si parla di dati provenienti da dispositivi IoT, incongruenza di alcune informazioni, ecc.

La regressione è una delle tecniche più utilizzate quando si parla di effettuare predizioni, ma essa è anche utilizzata per fare inferenza statistica, per testare ipotesi o per modellare relazioni di dipendenza.

Regressione lineare semplice

Nella regressione lineare, si assume che la variabile indipendente y sia esprimibile come combinazione lineare della variabile indipendente. Nella regressione lineare semplice, il modello che si cercherà di estrapolare dalle N osservazioni dei dati è:

$$y_i = \beta_0 + \beta_1 x_i + \varepsilon_i, \quad i = 1, \ldots\ldots, N$$

L'obiettivo di un modello di regressione è quello di ottimizzare i valori dei parametri β_0 e β_1, che rispettivamente rappresentano l'intercetta e l'inclinazione del modello. Diventa chiaro che un buon modello di regressione lineare ha il compito di individuare una retta che abbia la "pendenza giusta" e che riesca ad avvicinarsi al più possibile ai dati. ε_i rappresenta invece l'errore casuale in y corrispondente all'i-esima osservazione.

Nello specifico:

- β_0 corrisponde al valore medio di Y quando X = 0
- β_1 indica come varia Y in corrispondenza di una variazione unitaria di X

Esistono vari modi per valutare quanto il modello sia adatto ai dati. Il più comune è il metodo dei quadrati minimi. Questo metodo consiste nel rendere minima la somma dei quadrati delle differenze, tra i valori osservati (e quindi quelli reali) Y_{real} e i valori stimati (e quindi quelli predetti dal modello) $Y_{predict}$, quindi:

$$MSE = \sum_{1}^{N} (Y_{real} - Y_{predict})^2$$

Regressione lineare multipla

Più in generale, il modello di regressione lineare si estende alla presenza di più di una variabile indipendente. In questo modello ci sono p variabili indipendenti, e quindi:

$$y_i = \beta_0 + \beta_1 x_{1i} + \ldots + \beta_p x_{pi}\ \varepsilon_i, \quad i = 1, \ldots\ldots, N$$

Dal punto di vista teorico e anche da quello pratico, tutte le assunzioni fatte sul modello di regressione lineare semplice valgono per il modello di regressione lineare multipla, ovviamente, estendendo il calcolo dei residui (errori) ad ogni componente del modello.

Questo tipo di modello è quello che più conviene ai nostri task.

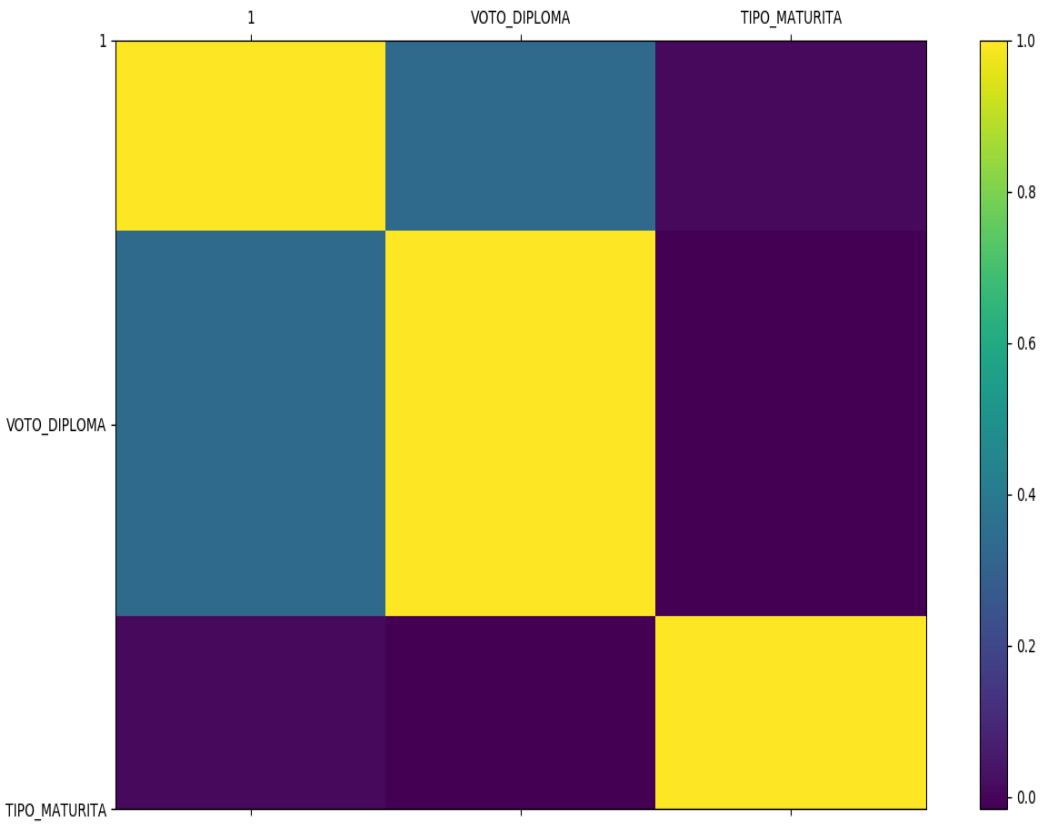
Correlation matrix

Prima di procedere con l'implementazione ci siamo chiesti se le variabili dipendenti che stavamo considerando riuscissero in qualche modo a rappresentare la realtà che volevamo catturare, ovvero, se il voto del diploma ed il tipo del diploma fossero in qualche modo dati correlati al numero di cfu che lo studente sostiene al primo anno, e quindi all'attributo "1" del nostro dataset.

La correlation matrix (matrice di correlazione) è appunto una matrice che mostra il coefficiente di correlazione tra variabili. Nel nostro caso, vogliamo capire quanto le variabili indipendenti siano correlate tra di loro e quanto le variabili indipendenti siano correlate alla variabile dipendente.

Il risultato che auspicavamo è che le variabili indipendenti siano il meno correlate possibile e che invece esse risultino essere molto correlate alla variabile dipendente.

L'indice di correlazione della correlation matrix è un numero reale che assume come valore massimo 1.00. Ovviamente, quanto più l'indice si avvicina ad 1.00 più le variabili sono correlate tra di loro. Analizzando la correlation matrix riguardante i nostri dati purtroppo ci siamo resi conto che tutti i presupposti per

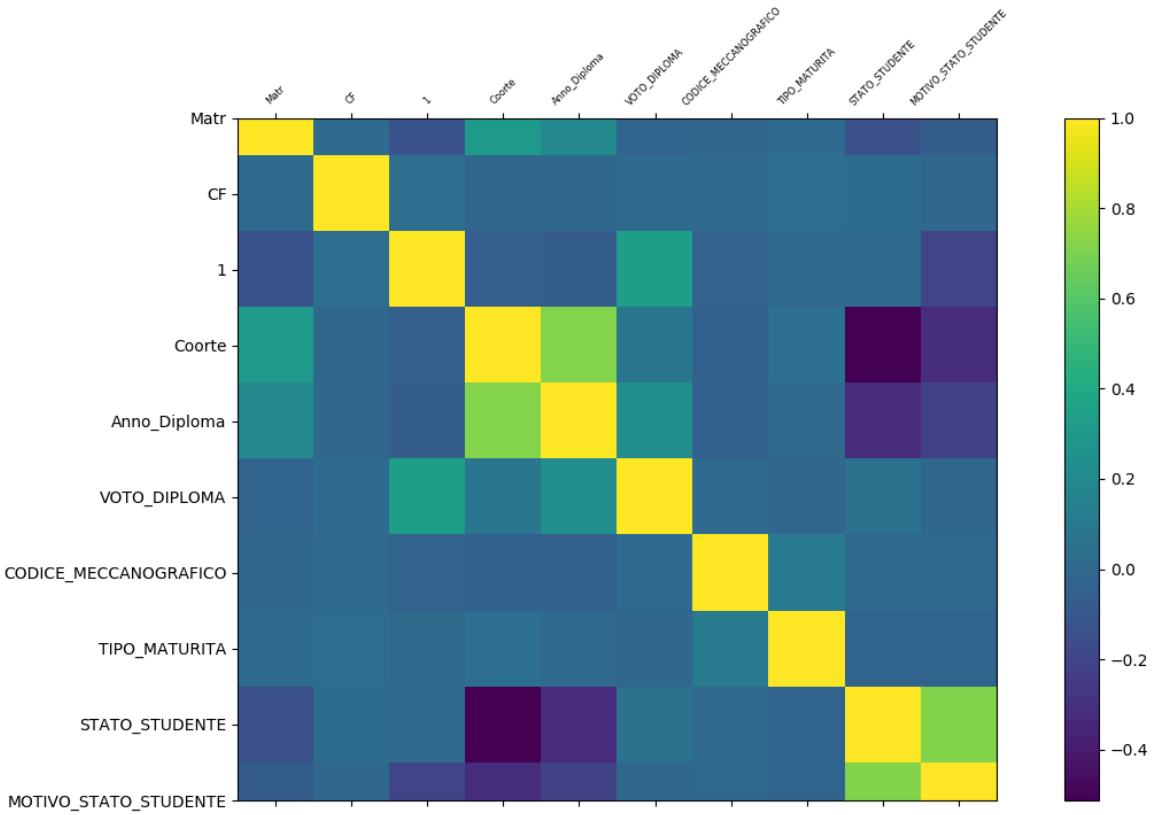


estrapolare un buon modello di regressione venivano meno, dato che le variabili indipendenti sono del tutto estranee alla variabile indipendente. Infatti:

è possibile notare il tutto dalla figura.

A questo punto, era evidente come utilizzando solo queste variabili indipendenti, il modello che saremmo riusciti ad estrapolare sarebbe stato altamente impreciso. Di conseguenza, abbiamo deciso di estendere il set delle variabili indipendenti a disposizione a tutti gli attributi che si possono avere in fase di immatricolazione dello studente (e quindi prima che esso sostenga alcun'esame), in modo da estendere le informazioni di cui eravamo a disposizione, ma non falsare eventuali analisi predittive.

A questo punto, abbiamo coinvolto altri attributi e ricalcolato la correlation matrix di questi:



è evidente, come la correlazione dei dati sia bassa anche in questo caso.

A questo punto, ci siamo chiesti se ci potesse essere una differenza di precisione utilizzando le variabili indipendenti richieste dal task, oppure utilizzando questo set di feature esteso.

Al fine di non precludere nessun risultato, abbiamo sviluppato tutti i modelli di regressione su entrambi i set di feature, analizzando in che modo si migliorasse o peggiorasse estendendo il set di feature.

R-squared

Per valutare la bontà dei modelli che andremo a misurare utilizzeremo l'R-squared come misura.

Questo score è una misura statistica che indica quanto il dato è vicino alla retta di regressione. È anche conosciuto come coefficiente di determinazione o coefficiente di multi-determinazione per la regressione multipla.

In pratica, misura la frazione della varianza della variabile dipendente espressa dalla regressione.

Si calcola in questo modo:

$$R^2 = \frac{ESS}{TSS} = 1 - \frac{RSS}{TSS}$$

dove:

- $ESS = \sum_{i=1}^{n} (\hat{y}_i - \overline{y})^2$, è la devianza spiegata dal modello
- $TSS = \sum_{i=1}^{n} (y_i - \overline{y})^2$, è la devianza totale
- $RSS = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$, è la devianza residua

e:

- y_i sono i dati osservati (quelli reali)
- $\overline{y}$ la media dei valori reali
- $\widehat{y_i}$ sono i dati che il modello sviluppato ha stimato

Di conseguenza R^2 varia tra 0 ed 1.

Tuttavia, quando si ha a che fare con modelli di regressione multipla si preferisce utilizzare un altro tipo di R-Squared che è l'Adjusted R^2

Adjusted R^2

È una variante dell' R^2 score ma utilizzato per l'analisi di regressione multipla. All'aumentare del numero di variabili indipendenti infatti aumenta anche il valore di R^2, per cui spesso viene utilizzato al suo posto $\overline{R^2}$. $\overline{R^2}$ può essere negativo e vale sempre la disuguaglianza $\overline{R^2} \leq R^2$.

$\overline{R^2}$ viene calcolato in questo modo:

$$\overline{R^2} = 1 - (\frac{RSS}{TSS}) \frac{n - 1}{n - k - 1}$$

dove:

- n è il numero delle osservazioni;

- k è il numero di variabili indipendenti.

Un buon modello dovrebbe avere il valore di questo indice quanto più prossimo ad 1, ma tuttavia quando si predice il comportamento di essere umani diventa quasi impossibile ottenere buoni risultati sotto questo punto di vista. Infatti, entrano in gioco molti fattori, che non possono essere soggetto di analisi statistica, come ad esempio la psicologia. In questi casi infatti, raramente si ottiene uno score superiore allo 0.5.

Linear regression

Questo modello è l'implementazione esatta della regressione lineare spiegata nei paragrafi precedenti.

Prima di procedere all'elenco dei risultati, è doveroso fare una piccola spiegazione di come sono stati suddivisi i dati in set di training e di test.

Quando si sviluppa un modello di ML, i dati che si hanno a disposizione, vengono suddivisi in due set, un set di training, che servirà ad allenare appunto il modello, ed un set di test, che servirà a valutare il modello.

Il nostro dataset non è un dataset di dati omogenei, quindi dividere il dataset con una tecnica totalmente random avrebbe influito negativamente sui risultati ottenuti. Di conseguenza, abbiamo suddiviso il dataset dapprima in due parti che comprendono gli studenti che si sono laureati e quelli che non si sono laureati.

Successivamente abbiamo composto il dataset di train e quello di test, mantenendo le stesse percentuali di composizione del dataset originario.

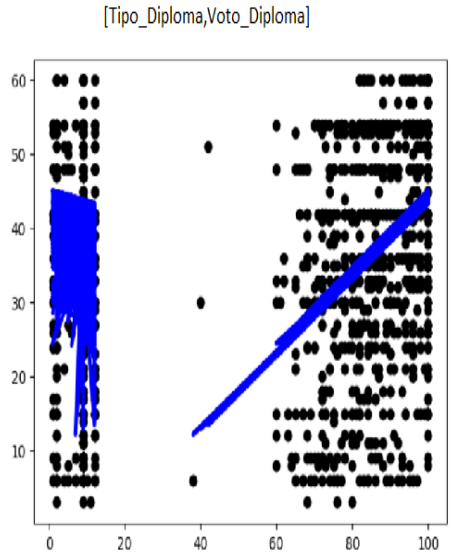
Ovviamente, lo splitting non avviene tagliando il dataset in due parti uguali. La maggior parte dei dati va affidata al dataset di train, di conseguenza, abbiamo composto il dataset di train con il 70% degli studenti laureati ed il 70% degli studenti non laureati. Il dataset di test quindi è stato composto con il 30% degli studenti laureati ed il 30% di studenti non laureati.

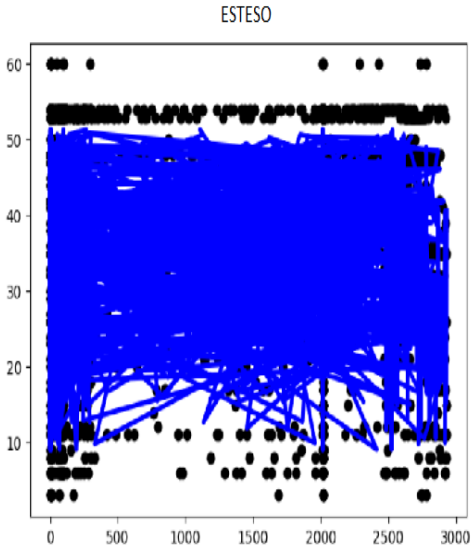
In definitiva, il nostro dataset di train è composto da un 70% del dataset originario e quindi comprende all'incirca 2900 tuple.

Come preaccennato, abbiamo sviluppato il modello su due set di feature diversi. Di seguito elenchiamo i risultati:

SET UTILIZZATO	SCORE (%)	MSE
[Tipo_Diploma,Voto_Diploma]	12.3	199.24
ESTESO	29.5	144.00

Come è possibile notare dai risultati, il set di feature esteso ha ottenuto uno score più alto e un errore più basso, ma tuttavia il modello non ha dato risultati soddisfacenti. Di seguito lasciamo il plotting dei due modelli:





è evidente, come entrambi i due modelli soffrano di unederfitting ed overfitting.

Polynomial regression

Questa tecnica effettua esattamente le stesse operazioni della regressione. Contrariamente alla regressione lineare però, non si sofferma a studiare i dati nel piano di dimensione due, ma effettua i suoi studi in un piano multidimensionale. L'idea è quella che magari qualche informazione potrebbe essere non visibile se studiata solo nel piano bidimensionale. Ecco i risultati del modello dopo aver adattato i dati in conformità a quanto richiesto:

SET UTILIZZATO	**SCORE (%)**	**MSE**
[Tipo_Diploma,Voto_Diploma]	1.5	190.90
ESTESO	1.5	133.04

Anche in questo caso i risultati non sono del tutto ottimali, ma l'incremento delle variabili indipendenti ha prodotto ancora una volta un effetto positivo abbassando notevolmente l'errore di predizione.

Ridge regression, elastic net, logistic regression e stochastic gradient descent

A questo punto abbiamo ad implementare questi quattro modelli, le quali puntano ad ottimizzare la valutazione dei modelli di regressione lineare.

Ecco i risultati:

Modello	**SET UTILIZZATO**	**Score (%)**	**MSE**	**SET UTILIZZATO**	**Score**	**MSE**
Elastic net	[Tipo_Diploma,Voto_Diploma]	3.4	206.31	ESTESO	28.4	146.82
Ridge Regression	[Tipo_Diploma,Voto_Diploma]	3.7	206.30	ESTESO	29.7	146.82
SGD Regression	[Tipo_Diploma,Voto_Diploma]	-∞	+∞	ESTESO	-∞	146.82
Logistic Regression	[Tipo_Diploma,Voto_Diploma]	10.5	534.41	ESTESO	8.4	146.82

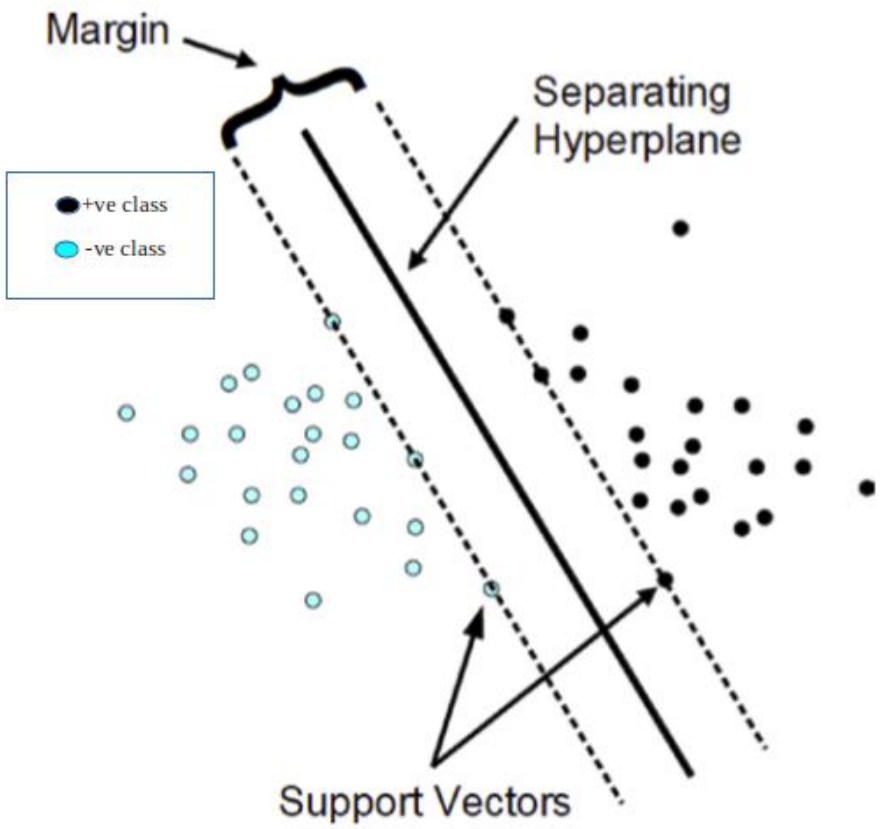
Ancora una volta, i risultati sono insoddisfacenti, ma si misura un miglioramento con il set di feature esteso

Support vector machine

Il support vector machine (SVM) è un algoritmo di apprendimento automatico supervisionato che può essere utilizzato sia per scopi di classificazione che di regressione. L'SVM è particolarmente adatto per la classificazione di set di dati complessi ma di piccole dimensioni. L'algoritmo si basa sull'idea di individuare un iperpiano che divida al meglio un set di dati in due classi. Nel caso di dati separabili linearmente in due dimensioni, un tipico algoritmo di apprendimento automatico tenta di trovare un limite che divide i dati in modo tale da ridurre al minimo l'errore di classificazione errata. SVM differisce dagli altri algoritmi di classificazione per il modo in cui sceglie **il limite di decisione** che massimizza la distanza dai punti dati più vicini di tutte le classi, ma non trova un limite di decisione qualsiasi, bensì quello.

Il limite di decisione più ottimale è quello che ha il margine massimo dai punti più vicini di tutte le classi. I punti più vicini dal limite di decisione che massimizzano la distanza tra il confine di decisione e i punti sono chiamati **vettori di supporto.**

Il limite di decisione in caso di macchine vettoriali di supporto è chiamato classificatore di margine massimo, o iperpiano di margine massimo.



In figura è rappresentato un esempio di utilizzo dell'algoritmo per effettuare una classificazione binaria.

Nel nostro caso abbiamo utilizzato SVM per fare regressione.

Support vector regression

La regression è una tecnica, come ampiamente descritto prima, che punta a stabilire una relazione funzionale tra variabili indipendenti e variabile dipendente. L'SVM è un algoritmo che lavora cercando uno o più iperpiani che dividono il

set di dati. Apparentemente l'SVM non è un algoritmo che si presta ad effettuare la regressione.

Per effettuare una regressione l'algoritmo cerca di minimizzare il valore normale del piano che individua. In pratica, l'algoritmo cerca di soddisfare la seguente condizione:

$$\forall n : |y_n - (x'_n\beta + b)| \leq \varepsilon$$

dove:

- y_n è l'osservazione reale della variabile dipendente
- x'_n è la variabile indipendente
- $x'_n\beta$ è la predizione effettuata della variabile dipendente
- ε è un valore di soglia specificato.

Non sempre diventa possibile individuare un iperpiano che soddisfi questo problema di minimo. Quello che si fa, è affidarsi a delle variabili di slack($\mu_n \; e \; \mu_n^*$), che introducono un valore di errore, necessario affinchè la condizione sia soddisfacibile. Il problema di minimo diventa quindi:

$$\forall n : |y_n - (x'_n\beta + b)| \leq \varepsilon + \mu_n$$

$$\forall n : |(x'_n\beta + b) - y_n| \leq \varepsilon + \mu_n^*$$

$$\forall n : \mu_n^* \geq 0$$

$$\forall n : \mu_n \geq 0$$

Ad ogni iterazione l'algoritmo quindi individua un iperpiano, e controlla se la predizione che ha fatto rispetta il vincolo che abbiamo specificato. Se il vincolo non viene rispettato all'algoritmo viene fornito come feedback una specie di "penalty", al contrario, se il vincolo è rispettato l'algoritmo continua nella sua esecuzione. Questo procedimento ha il compito anche di prevenire l'overfitting (ovvero il modello si adatta troppo a quelli che sono i dati).

Ovviamente, è possibile generalizzare il discorso, dalla regressione lineare semplice a quella multipla e ancora alla regressione polinomiale.

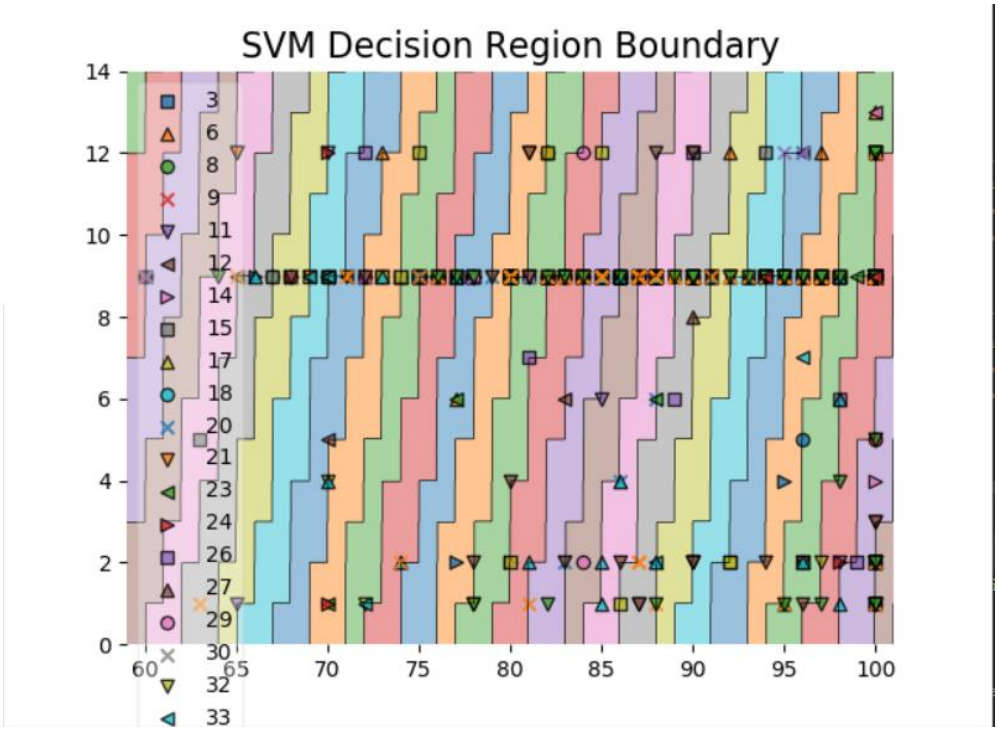
Svm linear regression per il task 1

L'algoritmo è stato usato per il task 1. Come al solito abbiamo lanciato l'algoritmo dapprima sul set di feature come da richiesta, e poi sul set di feature esteso.

I risultati sono sintetizzati nella tabella:

SET DI FEATURE	**SCORE (%)**	**MSE**
[TIPO_DIPLOMA, VOTO_DIPLOMA]	3.9	196.95
ESTESO	22.1	159.63

Come è possibile notare, anche l'SVM non ottiene dei risultati notevoli. D'altro canto, con il set di feature esteso si registra un notevole miglioramento sia dello score che dell'errore.



In figura sono riportati gli iperpiani individuati da SVM con il set di attributi semplice. Come si può notare, l'algoritmo ha una grande difficoltà nell'individuare un singolo iperpiano specifico, a conferma del fatto che i dati che avevamo a disposizione in questo set di feature erano poco performanti per il nostro obiettivo.

Decision tree

Un decision tree è un modello di apprendimento automatico supervisionato utilizzato per prevedere un obiettivo apprendendo le regole di decisione dalle funzionalità. Come già dal nome, l'idea è quella di pensare di scomporre i nostri dati prendendo decisioni sulla base di domande. Quindi sulla base delle funzionalità del nostro set di formazione, l'algoritmo apprende una serie di domande per dedurre le etichette di classe dei campioni. L'obiettivo è creare un modello che preveda il valore di una variabile target in base a diverse variabili di input. I Decision Tree si categorizzano rispetto alla variabile in output come: Categorical Decision Tree e Continuous Decision Tree.

Il modello è costruito dal partizionamento ricorsivo: a partire dal nodo radice, ogni nodo successivo può essere suddiviso in nodo figlio sinistro e destro che, a loro volta, possono essere suddivisi. Nasce spontanea la domanda, come facciamo a sapere qual è il punto di divisione ottimale in ciascun nodo?

A partire dalla radice, i dati vengono divisi in base ad una funzione. Iterativamente ripetiamo tale procedura per i nodi figli, finché le foglie sono pure, ovvero i campioni in ciascun nodo appartengono tutti alla stessa classe. Di solito si imposta un limite di profondità, poiché questo processo dettagliato potrebbe portare ad un eccesso di adattamento.

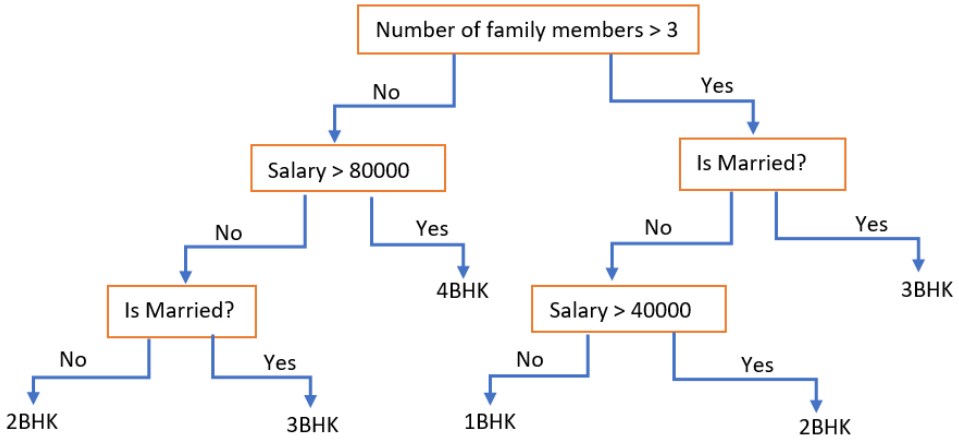
Al fine di dividere i nodi in base alle funzionalità più informative, dobbiamo definire una funzione obiettivo che vogliamo ottimizzare tramite l'algoritmo di apprendimento. Il criterio secondo il quale l'algoritmo divide in più rami i vari nodi dell'albero è critico per la precisone dell'algoritmo, differente poi se parliamo di ambito classificazione o regressione. In base ai criteri disponibili abbiamo delle metriche, queste metriche vengono applicate a ciascun sottoinsieme candidato e

i valori risultanti vengono combinati (ad esempio, media) per fornire una misura della qualità della divisione.

- Gini impurity
- Information-gain
- Variance reduction

L'unico svantaggio nell'utilizzare tali algoritmi è che sono proni all'overfitting, spesso è necessario settare dei vincoli o fare Pruring dei rami. Il motivo per il quale abbiamo optato nell'implementazione di tale modello è che l'algoritmo si presta bene quando abbiamo una relazione complessa tra gli attributi, una relazione che è difficile da spiegare infatti l'approccio non lineare del decison trre batte l'approccio lineare della regressione lineare.

In figura è rappresentato quello che dovrebbe essere l'output di un decision tree.

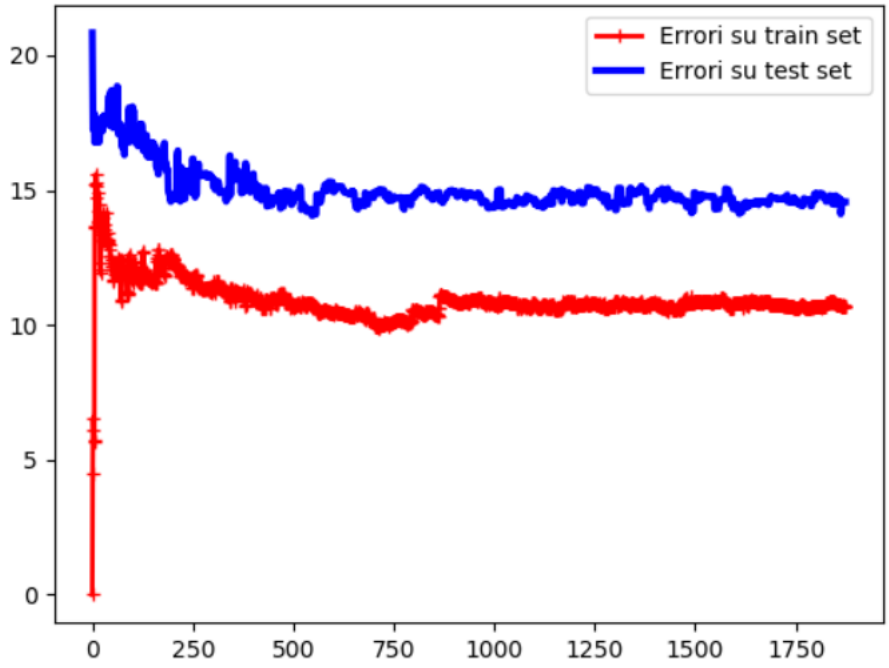


Decision tree regression per il task 1

I decision tree possono essere usati anche per un task di regressione. Quello che ci si chiede man mano che si scende di livello sull'albero è il valore della variabile dipendente. Di conseguenza, l'algoritmo cerca di ottimizzare una struttura ad albero nelle quali foglie ci si trovano range di valori che la variabile dipendente può assumere.

Applicando la tecnica al nostro task, i risultati che abbiamo ottenuti sono stati i seguenti:

SET DI FEATURE	**SCORE (%)**	**MSE**
[TIPO_DIPLOMA, VOTO_DIPLOMA]	5.4	193.88
ESTESO	17	170.01



In quest'immagine è riportato il tasso di errore che il decision tree, sul set di feature esteso, ottiene durante il procedimento di addestramento. Come è possibile notare man mano che cresce la dimensione del dataset l'entità dell'errore decresce. Questo grafico suggerirebbe che un aumento dei dati potrebbe portare ad una maggiore precisione delle predizioni effettuate.

La visualizzazione degli alberi è troppo complessa per essere riportata come tesi.

Nell'allegato 1, inseriremo le istruzioni per ottenere una visualizzazione web-based degli alberi.

Come in precedenza, anche questa volta, con il set di feature esteso si ottiene un notevole miglioramento sia dello score che dell'errore.

Random forest

Il random forest è una tecnica molto conosciuta e apprezzata dalla comunità scientifica.

Il concetto fondamentale alla base di random forest è semplice ma potente: "la saggezza della folla". È stato dimostrato scientificamente, che se poniamo la stessa domanda ad una folla di persone che non sono esperti dell'argomento, la risposta che ne verrà fuori sarà più precisa della risposta che potrebbe dare un esperto in materia.

Random Forest lavora proprio in questo modo. Anziché operare su di un solo albero di decisione specializzato, vengono utilizzati molteplici alberi di decisione (da qui il termine foresta) meno specializzati del singolo. L'obiettivo è alla fine quello di fare "bagging" (mettere insieme) sulle conoscenze ottenute.

Abbiamo applicato il random forest al task 1, ed i risultati ottenuti sono i seguenti:

seguenti:

SET DI FEATURE	SCORE (%)	MSE
[TIPO_DIPLOMA, VOTO_DIPLOMA]	7.7	189.08
ESTESO	41.3	120.37

Considerando, che come detto nella descrizione dell'indice di score, quando si ha a che fare con predizioni di comportamenti umani, questo score difficilmente supera lo 0.5, possiamo ritenerci soddisfatti di aver raggiunto uno score di 0.41 con il set di feature esteso. Se poi aggiungiamo il fatto che, l'errore è anche pressoché basso, possiamo concludere che per quanto riguarda la tecnica di regressione, RandomForest ottiene degli ottimi risultati.

Ci riserviamo, la possibilità di rieseguirlo su di un set di feature più consone ai nostri obiettivi.

Clustering

In statistica il clustering è un insieme di tecniche di analisi multivariata di dati volta al raggruppamento di elementi omogenei in un insieme di dati. Le tecniche di clustering si basano su misure relative alla somiglianza tra elementi, tale somiglianza concepita in termini di distanza in uno spazio multidimensionale.

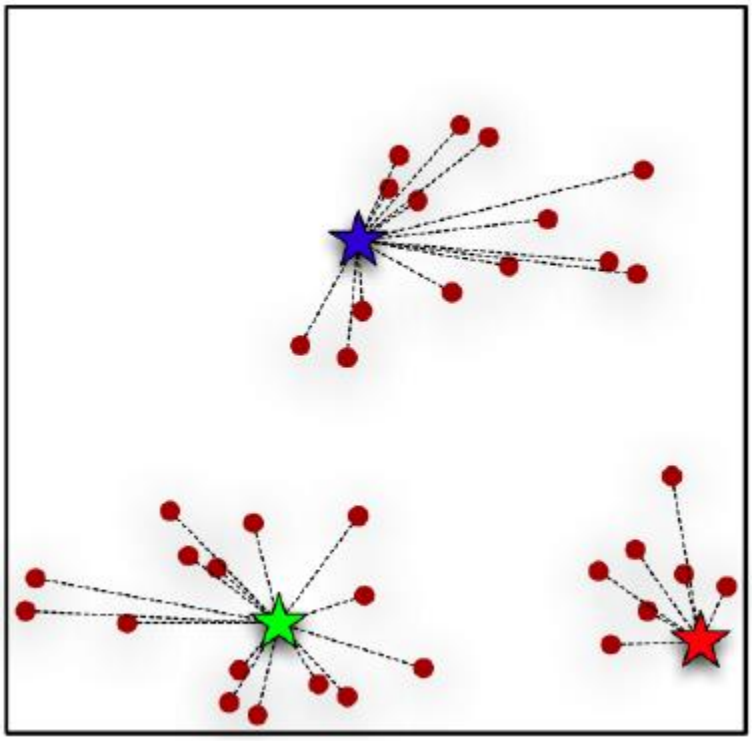
Per il nostro esperimento sono stati implementati i due algoritmi di clustering il K-means (unsupervisioned learning) e k-nearest neighbors - KNN - (supervisioned learning) più conosciuti e apprezzati.

Nei seguenti paragrafi verranno descritti in dettaglio i due algoritmi e sarà discussa la loro utilità per il raggiungimento del nostro obiettivo.

K-means

È un algoritmo di clustering partizionale, permette di suddividere un insieme di oggetti in k gruppi sulla base dei loro attributi. Si assume che gli attributi degli oggetti possano essere rappresentati come vettori e che, di conseguenza, formano uno spazio vettoriale. L'algoritmo ha come obiettivo principale quello di minimizzare la varianza totale intra-cluster. I cluster rappresentano le partizioni che dividono gli oggetti a seconda della presenza o meno di una certa somiglianza tra loro, il numero di cluster viene determinato a priori o con determinate tecniche, prima dell'esecuzione dell'algoritmo. Ognuno di questi cluster raggruppa un particolare insieme di oggetti, che vengono definiti data points. L'insieme dei data points analizzati definisce il set di dati, ovvero, l'insieme di tutte le istanze analizzate dall'algoritmo. Ogni cluster viene identificato mediante un centroide (il punto centrale).

Nella figura seguente i centroidi sono rappresentati dalle stelle blu, rosso, verde e i data points sono i puntini rossi vicino alle stelle.



Principio di funzionamento del k-means

L'algoritmo segue una procedura iterativa, inizialmente crea k partizioni e assegna ad ogni partizione i punti d'ingresso, casualmente o attraverso euristiche e calcola il centroide per ogni gruppo. L'algoritmo analizza ciascuno dei data points e li assegna al centroide più vicino, quindi viene calcolata la distanza euclidea tra ogni data points e ogni centroide. Ogni data points sarà assegnato al centroide la cui distanza risulti minima. Si ricalcolano poi i centroidi e si ricomincia finché

l'algoritmo non converge, ovvero, nessun data points cambia cluster, la somma delle distanze è ridotta al minimo o viene raggiunto un numero massimo prestabilito di iterazioni. L'algoritmo è molto adatto per scenari in cui è possibile creare gruppi di oggetti simili da una collezione di oggetti distribuiti casualmente. In generale, adatto in casi in cui conosci il numero di cluster a priori ed è possibile ottenere gruppi distinti da set di dati.

Il k-means ha il vantaggio di essere veloce, perché sono richiesti pochi calcoli (ovviamente ciò dipende dal set di dati e dal numero di cluster presenti). Lo svantaggio però è che non sempre conosci esattamente il numero di partizioni che vuoi visualizzare e soprattutto iniziando con una scelta casuale dei centroidi può produrre risultati di clustering diversi su diverse sequenze dell'algoritmo, pertanto potrebbe mostrare dati non reperibili e mancare di coerenza.

Utilizzo del k-means per il task1

Per loro natura gli algoritmi di clustering devono raggruppare i data point in insiemi di elementi tra di loro simili. Per quanto riguarda il Task 1, diventa evidente che i nostri data point possono essere molto distanti tra di loro, pur essendo simili in almeno due attributi su tre. Facciamo un esempio. Due studenti provenienti dallo stesso tipo di liceo e con due votazioni finali molto diverse tra di loro (uno studente totalizza 100 mentre l'altro 80), totalizzano al primo anno un numero di CFU molto simile tra di loro, ad esempio, lo studente che ha preso 100 totalizza 40 CFU, mentre quello che ha preso 80 ne totalizza 35. Questo inevitabilmente proietta questi due data point in cluster diversi tra di loro, pur essendo, alla luce dei risultati finali due studenti simili.

Per queste ragioni si è resa necessaria un'ulteriore fase di data preparation. In questa fase abbiamo rimodulato l'attributo "1". Sono stati create 4 classi, in questo modo:

1. **Classe 0:** da 0 a 15 CFU
2. **Classe 1:** da 16 a 30 CFU
3. **Classe 2:** da 31 a 45 CFU
4. **Classe 3:** da 45 CFU

Il criterio con cui sono state scelte le fasce di queste classi non è arbitrario. Principalmente avremmo voluto creare un numero maggiori di classi, basandoci su tutte le varie combinazioni di esami sostenuti. Chiariamo il concetto. Supponiamo che il primo anno sia composto dai seguenti esami:

1. Analisi Matematica – 9 CFU
2. Fisica – 12 CFU
3. Informatica – 6 CFU

La suddivisione ideale sarebbe stata quella rappresentata tra tutte le possibili combinazioni di questi 3 esami. Quindi:

1. **Cluster 0:** 6 CFU (INFORMATICA)
2. **Cluster 1:** 9 CFU (ANALISI MATEMATICA)
3. **Cluster 2:** 12 CFU (Fisica)

4. **Cluster 3:** 15 CFU (ANALISI MATEMATICA + INFORMATICA)
5. **…**
6. **Cluster i-esimo:** 27 CFU (ANALISI MATEMATICA + INFORMATICA + FISICA)

Tuttavia, utilizzare questo criterio sarebbe stato impossibile, dato che non eravamo a conoscenza dell'entità degli esami che gli studenti avrebbero dovuto sostenere, e pur avendo ricavato il piano di studi del corso di Laurea in questione, avremmo avuto il problema con gli studenti di anni precedenti (con ordinamenti e piani di studio differenti).

A questo punto la scelta più logica da fare è stata quella di individuare un modo per differenziare tra di loro gli studenti con numero di esami differenti. Abbiamo stabilito che una buona soglia di differenza è rappresentata da 15 CFU, che equivalgono ad un esame da 6 ed uno da 9 (che rappresentano la gran parte degli esami del primo anno). In conclusione, ecco come è stato possibile ricavare le classi che abbiamo descritto sopra.

Attenzione, la tecnica che abbiamo utilizzato non è empirica, ma tuttavia ci ha permesso di ottenere dei buoni risultati. Riteniamo che in un futuro lavoro sia possibile migliorare la suddivisione in classi di questo attributo.

A questo punto, abbiamo lanciato l'algoritmo del K-Means, chiedendo allo stesso la creazione di 4 cluster. I risultati verranno discussi approfonditamente con l'applicazione della logica Fuzzy, nel rispettivo paragrafo.

Utilizzo del k-means per il task2

Durante l'implementazione del k-means è stato subito evidente che il problema del dataset si ripresentava.

Come accennato nei precedenti capitoli il dataset in questione risultava essere molto incoerente ed eterogeneo, quindi, necessitava di un ulteriore lavoro di data preparation prima di avviare l'algoritmo

Durante questo sviluppo è emersa una nuova considerazione da fare, se prima ci siamo soffermati molto su ciò che richiedeva il primo task, adesso considerando il nuovo obiettivo è stato necessario applicare una nuova pulizia dei dati.

Facciamo un po' di chiarezza: Il secondo task ha come obiettivo quello di individuare un potenziale studente fuori corso, basandosi sulle informazioni riguardanti il suo background liceale e le sue performance nel primo anno di università, ossia rispettivamente, "Tipo_Diploma - Voto_Diploma" e "1" (numero di cfu sostenuti al primo anno). Analizzando i dati che avevamo a disposizione ci siamo resi conto che l'informazione "FC" subiva di forte inconsistenza. Infatti, per tutti gli studenti che non hanno completato il percorso di studi, o che hanno cessato la propria carriera comunque senza conseguire la laurea, l'informazione "FC" era settata a 0, lasciando intendere quindi erroneamente che lo studente non fosse andato fuori corso. A tutti gli effetti, tali studenti non sono andati fuori corso, ma data la scarsità di dati che avevamo a disposizione, abbiamo deciso di "punire" la rinuncia, la decadenza e la sospensione degli studi settando il rispettivo studente in fuori corso. In questo modo siamo riusciti a ricavare 778 tuple aggiuntive, che senza questa fase sarebbero andate perse, condizionando erroneamente qualsiasi lavoro di Machine Learning effettuabile.

Oltre a questa problematica, l'attributo "FC" era affetto anche dai problemi "canonici", ossia, mancanza del dato e incongruenza. Questi due ultimi problemi sono stati risulti facilmente, dato che avevamo a disposizione l'anno di immatricolazione e l'anno di laurea.

In questo modo il numero totale di studenti fuori corso sul quale abbiamo lavorato è salito da 898 a 1749.

A questo punto, composto il DataFrame dei dati necessari abbiamo lanciato il K-means.

All'algoritmo è stato chiesto di suddividere le tuple in due task, in modo da ricercare se ci fosse un modo netto di raggruppare gli studenti in corso e quelli fuori corso.

Sfortunatamente dall'analisi dei risultati è emerso che la disposizione degli studenti nel piano multidimensionale non è netta come ci auguravamo, bensì i dati non seguono nessun tipo di path, rendendo apparentemente impossibile la suddivisione.

Fuzzy k-means

L'osservazione che gli elementi di cui disponevamo erano molto eterogenei tra di loro ci ha portato ad adottare un approccio diverso alle tecniche di clustering.

Il clustering, come già spiegato, opera suddividendo gli elementi in insiemi di elementi simili tra di loro. Nel nostro caso, sebbene l'operazione sia possibile, non ci garantisce un alto grado di precisione.

Il Fuzzy Clustering è una tecnica adatta proprio in queste situazioni, ossia, quando non esistono dei veri e propri insiemi di elementi disgiunti tra di loro, bensì che si intersecano.

Questa tecnica infatti non si limita ad assegnare un elemento ad un insieme, ma ne stabilisce anche il grado di appartenenza.

Formalmente, sia X l'insieme dei data points. Un fuzzy set A è formato se esiste una funzione $f_A : X \rightarrow [0, 1]$ tale che ogni elemento $a \in$ A è della forma $f_A(x) = a$, per qualche $x \in$ X. Questo vuol dire che ad ogni data point in X è assegnato un valore compreso tra 0 e 1 che rappresenta il suo grado di appartenenza o la probabilità che questo elemento finisca proprio in questo insieme.

Nel Fuzzy Clustering quindi, un data point può essere assegnato ad uno o più cluster e il grado di appartenenza ad un cluster ne rappresenta la probabilità che esso sia proprio all'interno di quest'ultimo.

Fuzzy clustering per il task 1

Come si è dedotto dall'utilizzo del K-Means per il Task 1 i risultati ottenuti non soddisfacevano a pieno le nostre aspettative. Questo è dovuto al fatto che il dataset presenta elementi molto eterogenei tra di loro.

Il Fuzzy K-Means viene in aiuto in questi casi. Sebbene esistano diverse implementazioni consolidate del Fuzzy Clustering, come l'algoritmo C-Means e la versione Fuzzy del K-Means, abbiamo deciso di procedere ad una nuova implementazione basandoci su quelle che erano le nostre necessità, eliminando

così uno strato di complessità che avrebbe soltanto rallentato l'esecuzione, restituendoci risultati non del tutto consoni alle nostre domande.

Partendo dall'intuizione della tecnica Fuzzy, abbiamo deciso di lanciare l'algoritmo K-Means chiedendogli di suddividere i data points in quattro cluster. A questo punto ogni singolo studente del nostro dataset sarà etichettato con un intero compreso tra 0 e 4 che rappresenta il cluster in cui è stato collocato. I cluster appena creati non sono omogenei, bensì contengono al loro interno, potenzialmente elementi molto eterogenei tra di loro. Quello che abbiamo fatto è stato quello di contare all'interno di ogni cluster la presenza in percentuale delle diverse tipologie di studenti. Il risultato è il seguente:

	Classe 0 **0 – 15 CFU**	**Classe 1** **16 – 30 CFU**	**Classe 2** **31 – 45 CFU**	**Classe 3** **> 45 CFU**
Cluster 0 **[11, 88]**	10 %	20 %	33 %	37%
Cluster 1 **[11, 67]**	31 %	29 %	27 %	13 %
Cluster 2 **[11, 98]**	6 %	13 %	27 %	54 %
Cluster 3 **[11, 79]**	17 %	23 %	33 %	27 %

Questi risultati sono molto confortanti. La percentuale, indica la probabilità di appartenenza a quel cluster. Ad esempio, uno studente che al liceo ha preso un voto che si aggira intorno al 98 avrà il 54% di possibilità di sostenere più di 45 CFU, ma tuttavia avrà anche un 6% di possibilità di sostenerne meno di 15. Allo stesso modo uno studente che avrà ottenuto un voto più basso, avrà sì buone probabilità di sostenere pochi esami, ma il tutto dipende da altri fattori, e in definitiva non vuol dire che per forza il voto del diploma debba impattare sulle sue performance scolastiche.

Questi risultati, da una parte forniscono un ottimo feedback per quelli che sono gli obiettivi del lavoro e dall'altra confermano quanto evidenziato dallo stato dell'arte.

Ovviamente anche in questo caso, la scuola di provenienza converge verso lo scientifico, dato il peso specifico del dato presente all'interno del dataset.

Fuzzy clustering per il task 2

Nella nostra specifica implementazione, desideriamo come output il grado di appartenenza ad un solo cluster, che si possa tradurre quindi in percentuale secondo la quale lo studente rappresentato dal data point finisca fuori corso.

Il principio con cui abbiamo applicato la tecnica Fuzzy è a grandi linee quello utilizzato per il Task 1. Partendo dall'intuizione del Fuzzy clustering, abbiamo deciso di lanciare l'algoritmo K-Means, chiedendogli di suddividere i data points in due cluster. L'idea ancora una volta è quella di provare a suddividere nettamente in due insiemi gli studenti fuori corso e quelli in corso. Il K-Means ci restituirà il cluster di appartenenza per ogni Data Point.

A questo punto, avremo a disposizione due cluster 0 e 1, che senza perdita di generalità, supponiamo che rappresentino:

- Cluster 0: gli studenti che non sono andati fuori corso
- Cluster 1: gli studenti che sono andati fuori corso

Ovviamente, per quanto detto prima, questi due cluster non sono del tutto veritieri. Quindi a questo punto andremo a contare all'interno di ogni cluster quanti effettivamente sono gli studenti in fuori corso. Alla fine, calcoleremo la percentuale di studenti di fuori corso appartenenti al cluster preso in esame.

Questa percentuale sarà esattamente la percentuale che ha lo studente di andare fuori corso.

Analizziamo l'output per descrivere meglio questo concetto.

CLUSTER	**CENTROIDE**	**FUORI CORSO (%)**	**IN CORSO (%)**
0	[11, 72, 21]	96.4	3.6
1	[11,98,40]	65	35

Osservando questa tabella ci si può benissimo rendere conto che se uno studente viene classificato come appartenente al cluster 0 avrà il 3.2% di possibilità di non andare fuori corso, mentre se verrà classificato appartenente al cluster 1 avrà il 35% di possibilità di non andare fuori corso.

Questo risultato è ancora più interessante se si osservano i centroidi dei vari cluster.

Ricordiamo che la tupla che rappresenta il data point è [X, Y, Z], dove:

- X = Tipo di Diploma
- Y = Voto del Diploma
- Z = CFU sostenuti al primo anno

Nell'esempio è evidente come il tipo del diploma converga sempre ad 11, che rappresenta lo scientifico. Questo poiché come detto nei capitoli precedenti il liceo scientifico è di gran lunga il più presente all'interno del dataset. Ne consegue che l'attributo Tipo_Diploma avrà un peso specifico molto basso all'interno del nostro task di clusterizzazione.

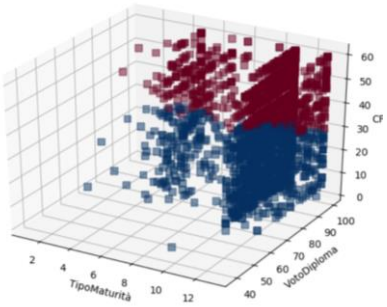
Successivamente, è interessante notare che gli studenti che hanno un voto al diploma più alto hanno molte meno probabilità di finire fuori corso rispetto a chi ha un voto più basso.

L'attributo che più incide però è ovviamente il numero di esami sostenuti al primo anno. Infatti, possiamo notare che i due cluster sono molto diversi sotto questo punto di vista.

Questo risultato ci ha soddisfatti, poiché pur essendo abbastanza impreciso nel determinare gli studenti in corso, risulta essere molto preciso nel determinare gli studenti fuori corso.

Ovviamente è molto più importante individuare questi ultimi che gli altri.

La seguente figura mostra la distribuzione dei data points nello spazio vettoriale:



K-Nearst Neighbors (KNN)

Il KNN è uno degli algoritmi più conosciuti nel machini learning, è un algoritmo di apprendimento supervisionato, il cui scopo è quello di predire una nuova istanza conoscendo i data points che sono separati in diverse classi. Dentro ad ogni classe vengono associati dei data points, il cui insieme definisce il set di dati. Vediamo un esempio:

- Iris, Giglio, Camelia... classe FIORI
- Abete, Olivo, Pino... classe ALBERI
- Adeina, Euforbia, Cactus... PIANTE GRASSE

L'algoritmo si basa sulla somiglianza delle caratteristiche, più in istanza è vicina a un data point, più il knn li considererà simili. Solitamente la distanza utilizzata è quella euclidea. Oltre alla distanza, l'algoritmo prevede di fissare un parametro k, scelto arbitrariamente, che identifica il numero di data points più vicini. L'algoritmo valuta le k minime distanze così ottenute. La classe che ottiene il maggior numero di queste distanze è scelta come previsione.

Il KNN è uno strumento non parametrico, non fa nessuna ipotesi sulla distribuzione dei dati che analizza. Questo significa che la struttura del modello è determinata dai dati ed è piuttosto utile, perché ne mondo reale, la maggior parte dei dati non obbedisce ai tipi assunti teorici fatti (come nei modelli di regressione lineare). Il KNN dovrebbe essere una delle scelte primarie per uno studio di classificazione quando c'è poca o nessuna conoscenza precedente sulla distribuzione dei dati.

La fase di addestramento è veloce, la mancanza di generalizzazione fa si che il KNN conservi tutti i dati di allenamento, ovvero quasi tutti i dati di addestramento sono necessari durante la fase di test.

Lo si usa a seconda del problema da risolvere considerando tre aspetti:

- Tipologia di problema: Classificazione, Regressione
- Tempo di calcolo: l'algoritmo consuma molta memoria
- Potere predittivo: quando k è piccolo stiamo limitando la previsione, l'algoritmo potrebbe essere cieco, un k troppo grande riduce l'impatto della varianza causato da un errore casuale, ma corre il rischio odi ignorare dettagli che potrebbero essere rilevati

L'Algoritmo funziona nel seguente modo, seleziona un valore K con cui prevedere il nuovo data point, ordina le distanze dalla più piccola alla più grande, e sceglie le prime k. Se è un problema di regressione si può restituire una media delle etichette K, se è classificazione sceglierà la classe che include più valori k trovati precedentemente. Facciamo un esempio, Supponiamo che vogliamo sapere la taglia di un cliente sulla base delle taglie raccolte di altri clienti, dove i parametri indipendenti sono altezza, decidiamo che il nostro K sia 5, l'algoritmo calcolerà le distanze euclidee, le ordinerà in maniera crescente e selezionerà le prime 5 istanze, quindi il valore della taglia di questi 5 clienti sarà la taglia predetta del nostro cliente.

A questo punto ci si chiede, ma come si sceglie k. In effetti la selezione del K è un compito cruciale per la buona riuscita della previsione, tra i vari metodi troviamo la CROSS VALIDATION. L'idea generale di questo metodo è quella

di dividere il campione di dati in un numero di k sottocampi ad astrazione casuale, disgiunti, che servono come fase di allenamento(training). Si applica poi il modello KNN per fare previsioni sul segmento k-esimo e si valuta l'errore.

In pratica, ogni volta un dei sottoinsiemi k viene usato come set di test, mentre gli altri k-1 sono messi insieme per formare un set di allenamento. Ogni data point può trovarsi in un set di test esattamente una volta, e può trovarsi in un set di allenamento k-1 volte.

Alla fine della procedura, gli errori calcolati vengono mediati per fornire una misura della stabilità del modello (quanto bene il modello prevede nuove istanze). I passaggi precedenti vengono quindi ripetuti per vari K e il valore che raggiunge l'errore più basso (o la massima precisione di classificazione) viene selezionando come valore ottimale per k(ottimale in senso di convalida incrociata). Si noti come la convalida incrociata sia dispendiosa dal punto di vista computazionale e si dovrebbe essere pronti a far funzionare per un po' di tempo, specialmente quando la dimensione del campione di esempi è ampia. In alternativa, puoi specificare un k ragionevole qualora si conosca il problema di analisi.

Il KNN è un algoritmo semplice non richiede formazione per fare prevision.

Uso del knn per il task2

Per l'implementazione del KNN abbiamo fatto riferimento al dataset costruito per l'implementazione del task2 dell'algoritmo K-means. L'obiettivo era quello di individuare se di un determinato studente fosse possibile predire il rischio di fuori corso sulla base dei k studenti più simili. Il nostro obiettivo è utilizzare il KNN per ricavare un'accuratezza alta.

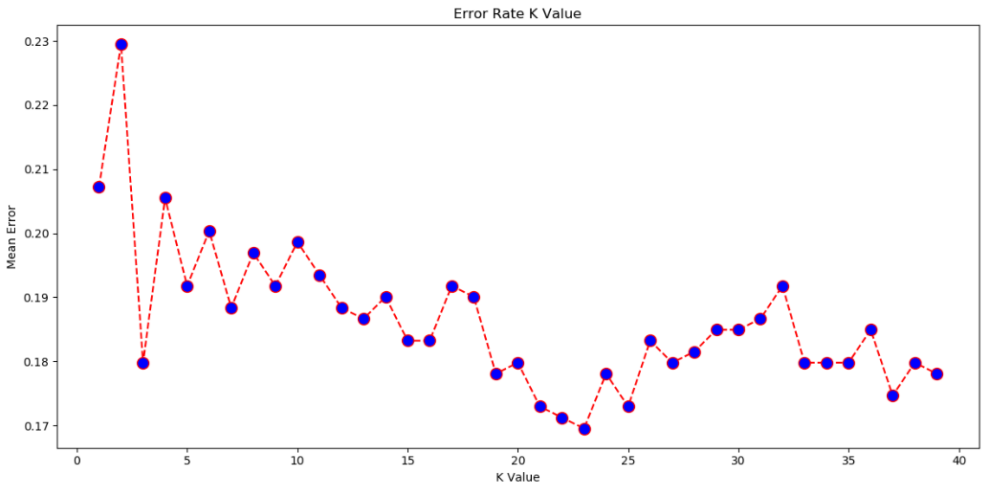
Le features che abbiamo considerato sono sempre le stesse, ovvero:

- I nostri attribuiti predittivi sono sempre Tipo maturità, Voto diploma, CFU1

- Il nostro attributo target è Fuori corso

Una volta selezionato il dataset preso atto dell'algoritmo abbiamo suddiviso il dataset in training set e test set.

Per l'individuazione del parametro K abbiamo tracciato il grafico del valore K e il tasso di errore corrispondente per il set di dati. La figura seguente mostra il grafico;



Una delle scelte empiriche seleziona k uguale al primo punto a gomito che si verifica dopo il valore 5, che è il numero di default dell'algoritmo.

Noi abbiamo deciso di lasciare K al valore di default (quindi k = 5), in quanto il valore della recall diminuiva considerando il k del primo gomito dopo 5. A questo punto non ci resta che lanciare il nostro KNN e prendere visione dei risultati.

	Precision	**Recall**	**F1-score**	**Support**
0	57%	46%	51%	117
1	87%	91%	89%	467

Ne segue che il grado di accuratezza nel predire i fuori corsi è dell'89%, mentre il grado di accuratezza nel predire gli in corso è del 57%. Una realtà che rispecchia i dati elaborati dal modello K-means.

Capitolo 5

Conclusioni

Predire comportamenti umani è una delle sfide più difficili nel contesto del Machine Learning.

Gli obiettivi di questo progetto erano quindi molto ambiziosi. Predire la carriera universitaria dello studente, poi, è ancora più complicato dato che entrano in gioco fattori come interazioni con altre persone e qualsiasi sollecitazione esterna al contesto scolastico ne può compromettere le performance.

Tuttavia, siamo riusciti ad ottenere dei risultati accettabili se considerata anche l'entità del dataset.

I risultati, e quindi la tecnica, più soddisfacente è stata sicuramente quella del Fuzzy K-Means, che ci ha permesso di stabilire con quali probabilità uno studente va fuori corso o meno, e con quale probabilità lo studente sostiene un certo numero di esami.

Da notare è il fatto che la precisione del Fuzzy K-Means per quanto riguarda il task 2 (Fuori Corso o meno) è più alta nell'individuazione del fuori corso. Questo è dovuto principalmente a due fattori:

1. Il dataset è caratterizzato da una maggioranza di studenti fuori corso
2. La sola conoscenza del background liceale e di quanti CFU sostenuti al primo anno, non garantisce una forte correlazione con l'andare o meno fuori corso.

Per il futuro sarebbe molto utile e stimolante continuare il lavoro, perfezionando le tecniche presentate in questa Tesi e raccogliendo le informazioni necessarie che sono state citate nello stato dell'arte.

Bibliografia e Referenze

- Normativa nazionale sistema universitario. http://www.miur.it/0006Menu_C/0012Docume/0098Normat/4640Modifi_cf2.htm

- A Review on Predicting Student's Performance Using Data Mining Techniques, Amirah Mohamed, Shahiri Wahidah, Husain Nur'aini, Abdul Rashid. [1]

- Z. Ibrahim, D. Rusli, Predicting students academic performance: comparing artificial neural network, decision tree and linear regression, in: 21st Annual SAS Malaysia Forum, 5th September, 2007. [2]

- D. M. D. Angeline, Association rule generation for student performance analysis using apriori algorithm, The SIJ Transactions on Computer Science Engineering & its Applications (CSEA) 1 (1) (2013) p12-16. [3]

- M. M. Quadri, N. Kalyankar, Drop out feature of student data for academic performance using decision tree techniques, Global Journal of Computer Science and Technology 10. [4]

- E. Osmanbegović, M. Suljić, Data mining approach for predicting student performance, Economic Review 10. [5]

- W. Hämäläinen, M. Vinni, Comparison of machine learning methods for intelligent tutoring systems, in: Intelligent Tutoring Systems, Springer, 2006, pp. 525-534. [6]

- M. M. A. Tair, A.M. El-Halees, Mining educational data to improve students performance: a case study, International Journal of Information 2. [7]

- M. Mayilvaganan, D. Kalpanadevi, Comparison of classification techniques for predicting the performance of students academic environment, in: Communication and Network Technologies (ICCNT), 2014 International Conference on, IEEE, 2014, pp. 113-118. [8]

- S. Natek, M. Zwilling. Student data mining solution–knowledge management system related to higher education institutions. Expert systems with applications, 41 (14) (2014). [9]

- T. M. Christian, M. Ayub, Exploration of classification using nbtree for predicting students' performance, in: Data and Software Engineering (ICODSE), 2014 International Conference on, IEEE, 2014, pp. 1-6. [10]

- K. F. Li, D. Rusk, F. Song, Predicting student academic performance, in: Complex, Intelligent, and Software Intensive Systems (CISIS), 2013 Seventh International Conference on, IEEE, 2013, pp. 27-33. [11]

- U. bin Mat, N. Buniyamin, P.M. Arsad, R. Kassim, An overview of using academic analytics to predict and improve students' achievement: A proposed proactive intelligent intervention, in: Engineering Education (ICEED), 2013 IEEE 5th Conference on, IEEE, 2013. [12]

- G. Elakia, N.J. Aarthi, Application of data mining in educational database for predicting behavioural patterns of the students, Elakia et al,/(IJCSIT) International Journal of Computer Science and Information Technologies 5. [13]

- V. Oladokun, A. Adebanjo, O. Charles-Owaba. Predicting students academic performance using artificial neural network: A case study of an engineering course. The Pacific Journal of Science and Technology, 9 [14]

- V. Ramesh, P. Parkavi, K. Ramar. Predicting student performance: a statistical and data mining approach. International Journal of Computer Applications, 63. [15]

- T. Mishra, D. Kumar, S. Gupta, Mining students' data for prediction performance, in: Proceedings of the 2014 Fourth International Conference on Advanced Computing & Communication Technologies, ACCT '14, IEEE Computer Society, Washington, DC, USA, 2014. [16]

- S. Sembiring, M. Zarlis, D. Hartama, S. Ramliana, E. Wani. Prediction of student academic performance by an application of data mining techniques, in: International Conference on Management and Artificial Intelligence IPEDR, 6 (2011). [17]

- G. Gray, C. McGuinness, P. Owende, An application of classification models to predict learner progression in tertiary education, in: Advance Computing Conference (IACC), 2014 IEEE International, IEEE, 2014. [18]

- I. Hidayah, A.E. Permanasari, N. Ratwastuti, Student classification for academic performance prediction using neuro fuzzy in a conventional classroom, in: Information Technology and Electrical Engineering (ICITEE), 2013 International Conference on, IEEE, 2013. [19]

- S. S. Meit, N.J. Borges, B.A. Cubic, H.R. Seibel, Personality differences in incoming male and female medical students., Online Submission. [20]