Master in Data Science for Economics, Business and Finance

Prof. Alfio Ferrara

Corso di Programmazione Python

Tracce di progetto per la prova finale

Introduzione e modalità di svolgimento della prova

La prova finale del corso consta della discussione di un progetto d'esame svolto da un gruppo di studenti di numerosità variabile fra un **minimo di 1 persona** e un **massimo di 3 persone**.

La scelta della traccia di progetto è completamente libera. Le seguenti tracce vanno intese come spunti e suggerimenti, ma possono essere liberamente modificate o anche sostituite con idee progettuali diverse, purché concordate col docente.

Modalità di valutazione: la valutazione dei risultati è individuale e si basa su una presentazione dei risultati da parte degli autori del progetto. La presentazione va illustrata con lucidi e deve durare 10 minuti a cui fanno seguito 10 minuti di domande e discussione fra docente e studenti, come in un workshop scientifico o aziendale.

Criteri di valutazione

- capacità di sintetizzare i risultati progettuali in modo efficace nel tempo dato
- estensione e completezza dell'attività di progetto
- padronanza delle metodologie e delle competenze di programmazione utilizzate per il progetto
- creatività e capacità di sostenere le proprie scelte

Nota: Le tracce di progetto definiscono un problema ma non forniscono volutamente indicazioni sulle metodologie o sulle tecniche di programmazione più adeguate per risolverlo. Parte integrante della valutazione è verificare la capacità degli studenti di individuare le metodologie e gli strumenti che ritengono più efficaci per risolvere il quesito. La capacità di documentarsi e la creatività sono parte integrante delle competenze oggetto di verifica.

Modalità di consegna

Il codice sorgente del programma oggetto della consegna, la documentazione e la presentazione vanno caricate su un apposito repository `GitHub` creato dagli autori del progetto che devono poi darne accesso al docente.

Importante: il progetto deve includere un file `README` di presentazione e un notebook `main.ipynb` in cui, opprtunamente documentato, risieda il codice necessario a avviare e eseguire il progetto stesso. Il programma deve essere eseguibile e i risultati riproducibili.

Progetti

P1: Lascia o raddoppia

Creare quiz a risposta multipla, in cui una sola delle risposte possibili a un quesito è quella corretta è più difficile di quanto si possa pensare. Il rischio infatti è di creare quesiti troppo semplici o, al contrario, quesiti troppo difficili.

Lo scopo del progetto è di creare automaticamente quiz composti da questiti sul cinema, sfruttando i dati di IMDb. Ogni quesito deve essere composto da una domanda e quattro possibili risposte, di cui una sola corretta. Il programma che genera i quesiti deve anche implementare un criterio che permetta di generare le risposte possibili in modo proporzionale alla difficoltà del quesito desiderato. Il programma deve inoltre permettere a un giocatore umano di rispondere al quiz e deve misurarne la prestazione con un punteggio complessivo che tenga in considerazione la difficoltà di ogni singolo quesito.

I dati di IMDb possono essere acquisiti tramite specifiche API (come ad esempio IMDbPY) o dataset esistenti, come ad esempio IMDb Dataset.

P2: Sogni d'oro

Il dataset scelto per il progetto, disponibile all'indirizzo hotels, contiene in forma sintetica le seguenti informazioni:

- `hotels.xlsx`: per 400 hotel è disponibile l'informazione circa il numero di camere libere e il costo unitario di ogni camera
- `guests.xlsx`: per 4000 clienti potenziali è disponibile l'informazione sulla frazione del costo unitario della camera a cui hanno diritto come sconto
- `preferences.xlsx`: per ogni cliente è disponbile l'informazione sull'ordine di preferenza dell'hotel, dove il valore numerico associato alla preferenza indica l'ordine di preferenza (i.e., 1 è la prima scelta, 2 la seconda, etc.).

Il programma deve calcolare l'allocazione dei clienti presso gli hotel, considerando il numero di stanze disponibili, il fatto che ogni cliente occupa esattamente una camera e che ogni pernottamento dura una sola notte. Il prezzo pagato dal cliente è il prezzo unitario della stanza scontato della frazione di sconto a cui il cliente ha diritto.

Il programma deve realizzare quattro diverse strategie di allocazione:

- Casuale: i clienti sono distribuiti casualmente nelle stanze fino a esaurimento dei posti o dei clienti
- Preferenza cliente: i clienti sono distribuiti nelle stanze in ordine di prenotazione (il numero del cleinte ne indica l'ordine) e attribuendogli l'hotel in ordine di preferenza, fino a esaurimento dei posti o dei clienti
- Prezzo: i posti in hotel sono distribuiti in ordine di prezzo, partendo dall'hotel più economico e in subordine in ordine di prenotazione e preferenza fino a esautimento posti o clienti
- Disponibilità: i posti in hotel sono distribuiti in ordine di disponibilità di stanze, partendo dall'hotel più capiente e in subordine in ordine di prenotazione e preferenza fino a esautimento posti o clienti

Il programma deve infine presentare e visualizzare un report del risultato ottenuto, riportando per ogni strategia il numero di clienti sistemati, il numero di stanze occupate, il numero di hotel diversi occupati, il volume complessivo di affari (guadagno complessivo di ogni hotel) e il grado di soddisfazione dei clienti (calcolato in funzione della posizione dell'hotel loro assegnato rispetto alle loro preferenze).

P3: Il migliore di tutti

Il sito BoardGameGeek (BGG) fornisce dati e statistiche di vario genere sull'entusiasmente hobby del gioco da tavolo. Gli utenti di BGG commentano i giochi e opzionalmente assegnano loro un punteggio di gradimento compreso fra 0 e 10.

Il dataset del progetto, disponibile al link bggmastercobra contiene circa 10^6 commenti per circa 30 000 giochi. Lo scopo del progetto è produrre un ranking dei giochi in ordine di gradimento degli utenti. Si noti che per ottenere un ranking adeguato occorre considerare non solo i voti, ma anche il fatto che i diversi giochi possono essere associati a un numero molto diverso di commenti e quindi di voti di singoli utenti. Per una discussione su questo punto si veda ad esempio How Not To Sort By Average Rating.

Il progetto dovrà proporre una propria startegia di ordinamento dei giochi, sostenendone l'adeguatezza anche in relazione al ranking ufficiale di BGG, disponibile online link.

Il progetto dovrà inoltre produrre un report e dei grafici che confrontino il ranking prodotto con il ranking ottenuto attraverso la sola media voto di ogni gioco, mostrando la diversa distribuzione dei punteggi e le differenze riscontrabili nel ranking finale.

P4: Guarda se fa freddo

Il dataset GlobalLandTemperaturesByMajorCity.csv riporta la temperatura registrata nelle maggiori città del mondo a partire dal 1750. Usando questi dati, il progetto dovrà fornire una visualizzazione grafica efficace del variare delle temperature nel tempo, evidenziando le città in cui si sono registrate le maggiori escursioni termiche nei diversi periodi storici.

Per la visualizzazione dei dati su mappa si veda geopandas.

Il programma inoltre dovrà suggerire, a seconda del periodo considerato, il percorso migliore da seguire per un viaggatore freddoloso che intenda spostarsi da Pechino a Los Angeles muovendosi tappa dopo tappa verso la città più calda fra le 3 a lui più vicine.

P5: Avere sempre la risposta giusta

CISI è un dataset che permette di valutare le performance di un sistema di Information Retrieval. In particolare, CISI offre una serie di documenti testuali e di query, riportando la rilevanza dei diversi documenti per ogni query. il progetto ha lo scopo di usare tale dataset per valutare le performance di un sistema di ricerca di documenti basato su query in linguaggio naturale.

Si richiede pertanto di costruire tre semplici motori di ricerca: il primo basato su risposte casuali alle interrogazioni; il secondo basato sulla frequenza dei termini nei documenti; il terzo basato sulla misura di rilevanza dei termini nei documenti calcolata con Tfldf.

Usando i dati di CISI si richiede di formulare una strategia di valutazione dei tre motori di ricerca e di produrre un report che valuti comparativamente i tre sistemi, fornendo un'indicazione delle cause di errore più rilevanti per ognuna delle tre alternative.

P6: I mammiferi depongono uova?

I dataset Zoo fornisce una serie di dati relativi a diverse specie animali al fine di classificarle in 7 diverse classi, ovvero mammiferi, uccelli, rettili, pesci, anfibi, insetti e invertebrati.

Seguendo un approccio non supervisionato, ovvero senza osservare la classe di ogni specie animale, il progetto mira a confrontare le diverse specie e raggruparle utilizzando diversi algoritmi di clustering.

Confrontando poi il risultato di ogni algoritmo, si intende mostrare quale algoritmo di clustering approssimi meglio le classi fornite dal dataset.

Si richiede pertanto non solo di definire una metodologia per confrontare i risulati del clustering con la classificazione attesa, ma anche di descrivere in modo sintetico le caratteristiche distintive di ciascun cluster di specie prodotto dall'algoritmo oggetto della valutazione.

P7: Il giro del mondo in 80 giorni

Si consideri l'insieme dei dati che descrivono alcune delle principali città del mondo (link). Si supponga che sia sempre possibile viaggiare da ogni città fino alle 3 città più vicine e che tale viaggio richieda 2 ore per la città più vicina, 4 ore per la seconda città più vicina e 8 ore per la terza più vicina. Inoltre il viaggio richiede 2 ore aggiuntive se la città di destinazione è in un'altra nazione rispetto alla città di partenza e altre 2 ore aggiuntive se la città di destinazione ha più di 200 000 abitanti.

Partendo la Londra e viaggiando sempre verso Est, è possible compiere il giro del mondo tornando a Londra in 80 giorni? Quanto tempo si impiega al minimo?