D.A.

1. Quali sono i tipi principali di dati e di analisi nel campo della data analytics?

Nel campo della data analytics, i dati possono essere classificati in tre tipologie principali: Strutturati, Semi-Strutturati e Non Strutturati. A seconda della natura dei dati e degli obiettivi, si possono effettuare diverse tipologie di analisi:

- Descrittiva: Risponde alla domanda "cos'è successo?", fornendo un riassunto delle tendenze e dei pattern nei dati.
- Diagnostica: Risponde alla domanda "perché è successo?", esplorando le cause dietro gli eventi osservati.
- Predittiva: Risponde alla domanda "cosa succederà?", utilizzando modelli per prevedere eventi futuri basandosi sui dati storici.
- Prescrittiva: Risponde alla domanda "cosa possiamo fare affinché qualcosa accada?", suggerendo azioni specifiche per influenzare i risultati desiderati.

2. Perché il pre-processing dei dati è cruciale e quali sono le sue strategie principali?

Il pre-processing dei dati è cruciale perché i dati del mondo reale sono spesso incompleti, rumorosi e inconsistenti, rendendo impossibile ottenere risultati di qualità senza un'adeguata preparazione. Le principali strategie per il miglioramento dei dati includono:

- Data Cleaning: Gestisce dati mancanti (MCAR, MAR, NMAR) tramite ignoranza, conversione in valori noti o imputazione basata su strategie come "Most Common" o K-NN. Elimina il rumore (ad esempio, con la discretizzazione) e identifica/rimuove gli outlier.
- Unbalanced Data: Affronta i dataset con classi sbilanciate utilizzando Oversampling (es. SMOTE per generare istanze sintetiche della classe minoritaria) o Undersampling (es. Tomek Link Method per rimuovere istanze della classe maggioritaria che non formano legami con la minoritaria).
- Data Reduction: Riduce il numero di feature per migliorare l'efficienza del modello senza perdere informazioni significative. Questo può avvenire tramite Feature Extraction (mappando le feature in uno spazio di dimensione inferiore) o Feature Selection (selezionando un sottoinsieme delle feature originali tramite FILTERS o WRAPPER).
- Data Transformation e Data Integration sono altre fasi importanti per preparare i dati.

3. Cosa sono le reti e quali proprietà e misurazioni vengono utilizzate per descriverle?

Le reti sono modelli matematici composti da nodi (che rappresentano entità) e archi (che rappresentano le connessioni tra i nodi). Possono essere connesse, non connesse (formando componenti separate) o alberi (connesse e prive di cicli). Per descrivere le reti, specialmente quelle di grandi dimensioni, si

utilizzano diverse proprietà e misurazioni:

- Grado dei nodi: L'Indegree (archi in entrata) e l'Outdegree (archi in uscita) di un nodo. La loro somma è il grado totale. Aiuta a identificare gli "hub" della rete (nodi con grado elevato).
- Paths e Distanza: Il shortest path (percorso più breve tra due nodi). Da questo si calcolano la distanza media della rete e il diametro (distanza massima).
- Coefficiente di Clustering: Indica la probabilità che i vicini di un nodo siano tra loro collegati, misurando la coesione locale.
- Centralità: Definisce l'importanza di un nodo in base a vari criteri:
  - Degree Centrality: Quanti collegamenti diretti ha un nodo.
  - Betweenness Centrality: Quante volte un nodo funge da ponte nei percorsi più brevi tra altri nodi.
  - Closeness Centrality: Quanto velocemente un nodo può raggiungere tutti gli altri.
  - Eigenvector Centrality: Quanto un nodo è collegato a nodi importanti.
    - PageRank Centrality: Stima l'importanza di un nodo in un grafo diretto basandosi sulla rilevanza dei nodi che puntano ad esso.
- Reciprocità: Il numero di relazioni reciproche rispetto al totale.
- Densità: Il numero di archi rispetto al totale dei nodi possibili, indicando quanto è ben connessa una rete.

4. Cos'è la Community Detection e quali approcci esistono per identificare le community?

La Community Detection è il processo di identificare gruppi di nodi all'interno di una rete che sono più densamente collegati tra loro rispetto al resto della rete. L'identificazione delle community è fondamentale per scoprire relazioni nascoste, comprendere le interazioni, inferire valori mancanti e prevedere collegamenti. Esistono diversi approcci per definire e raggruppare le community:

- Node-Centric: I nodi devono soddisfare requisiti specifici individualmente per appartenere a una community.
- Group-Centric: Un dato gruppo di nodi deve soddisfare determinati requisiti collettivi.
- Network-Centric: La rete viene partizionata in insiemi disgiunti. Questo include tecniche basate sulla similarità dei nodi (es. vector similarity), algoritmi di clustering come k-means o spectral clustering, e la Modularità Maximization, che cerca di massimizzare la densità di archi all'interno delle community e minimizzare quelli tra community diverse.
- Hierarchy-Centric: Si definiscono strutture gerarchiche basate sulla topologia della rete, spesso rimuovendo ricorsivamente gli archi con la betweenness centrality più alta per rivelare partizionamenti a diversi livelli.

5. Che cos'è il Natural Language Processing (NLP) e come viene rappresentato il testo per l'analisi computazionale?

Il Natural Language Processing (NLP) è un campo dell'informatica che si occupa di permettere ai computer di comprendere, interpretare e generare il linguaggio umano. L'NLP è complesso a causa dell'ambiguità del linguaggio, del contesto implicito, della variabilità linguistica (sinonimi, stili) e della sua struttura grammaticale complessa. Per essere elaborato dai modelli di NLP, il testo deve essere convertito in un formato numerico. Le principali tecniche di rappresentazione del testo sono:

- Bag of Words (BoW): Rappresenta ogni documento come un conteggio delle occorrenze di ogni parola. Perde l'ordine e il contesto delle parole.
- Term Frequency-Inverse Document Frequency (TF-IDF): Assegna un peso alle parole basato sulla loro frequenza nel documento specifico e sulla loro rarità nell'intero corpus, dando maggiore rilevanza a parole distintive.
- Word Embeddings (es. Word2Vec, GloVe): Trasformano le parole in vettori densi che catturano il loro significato e le relazioni semantiche (es. analogie come "re - uomo + donna ≈ regina").
- Tokenizzazione + Transformers (es. BERT, GPT): Dividono il testo in "token" (parole, subwords o caratteri) e li convertono in embedding contestuali, dove il significato di un token dipende dalle parole circostanti, catturando relazioni complesse e a lungo raggio.

6. Cosa sono i Word Embeddings e come funzionano Word2Vec e le tecniche di riduzione di dimensionalità per la loro visualizzazione?

I Word Embeddings sono rappresentazioni vettoriali dense delle parole che catturano le loro relazioni semantiche e sintattiche. Sono generati da modelli che imparano il significato delle parole dal loro contesto. Word2Vec è un modello popolare che crea questi embedding. Funziona in due architetture principali:

- CBOW (Continuous Bag of Words): Predice la parola centrale dato il contesto (parole circostanti).
- Skip-Gram: Predice le parole di contesto data una parola centrale. Entrambe le architetture usano una semplice rete neurale che mappa una parola (in input) in un vettore denso (embedding), cercando di prevedere il suo contesto o la parola stessa.

I vettori risultanti consentono analogie come "re - uomo + donna ≈ regina". Dato che gli embedding hanno spesso un'alta dimensionalità, per la loro visualizzazione è necessario ridurla a 2 o 3 dimensioni.

Le tecniche comuni di Dimensionality Reduction includono:

- Principal Component Analysis (PCA): Una tecnica lineare che identifica le direzioni di massima varianza nei dati per proiettarli in uno spazio a dimensioni inferiori. È veloce ma può perdere informazioni e non cattura relazioni non lineari.
- t-distributed Stochastic Neighbor Embedding (t-SNE): Una tecnica non lineare che eccelle nel visualizzare strutture complesse preservando le relazioni locali tra i punti. È computazionalmente più costosa e non produce una trasformazione riutilizzabile su nuovi dati.

7. Cosa sono i Large Language Models (LLM) e quali sono le differenze chiave tra BERT e GPT?

I Large Language Models (LLM) sono modelli di intelligenza artificiale avanzati che generano "contextual embeddings", cioè rappresentazioni vettoriali delle parole che tengono conto dell'intero contesto in cui si trovano. Questo supera la limitazione dei word embeddings tradizionali che producono una singola rappresentazione per parola, indipendentemente dal contesto. I LLM, in particolare quelli basati su architetture Transformer, sono fondamentali per la comprensione e generazione del linguaggio.

Le due famiglie di LLM più influenti sono BERT (Bidirectional Encoder Representations from Transformers) e GPT (Generative Pre-trained Transformer):

AspettoBERT (Encoder)GPT (Decoder)ArchitetturaEncoder-onlyDecoder-onlyDirezioneBidirezionale (usa contesto completo)Autoregressivo (da sinistra a destra)TrainingMasked Language Modeling (MLM)Causal Language ModelingOutputEmbedding contestualizzatiToken successivi + probabilitàTask idealiClassificazione, QA, NER, riassuntoGenerazione testo, traduzione, completamentoBERT è ottimizzato per la comprensione del testo, prevedendo parole mascherate usando sia il contesto sinistro che quello destro. GPT è ottimizzato per la generazione di testo, prevedendo il prossimo token in modo autoregressivo. Implementa anche fasi di fine-tuning e Reinforcement Learning from Human Feedback (RLHF). Modelli come LLaMA sono sviluppi recenti dei Transformer con ottimizzazioni specifiche come le funzioni di attivazione SwiGLU, gli Embedding Posizionali Rotatori (RoPE) e la normalizzazione RMSNorm, migliorando efficienza e capacità.

8. Cos'è l'Explainability nei LLM e quali tecniche locali vengono utilizzate per spiegare le loro predizioni?

L'Explainability nei Large Language Models (LLM) è la capacità di spiegare il comportamento e le predizioni di un modello in termini comprensibili all'uomo. È fondamentale per costruire fiducia negli utenti e per aiutare i ricercatori a identificare bias, rischi e limiti dei modelli, promuovendone il miglioramento. Esistono diversi paradigmi di explainability, tra cui il Traditional Fine-Tuning, il Prompting e l'Evaluation delle spiegazioni stesse.

Per la spiegazione locale, che mira a giustificare una singola predizione specifica, le tecniche principali sono basate sulla Feature Attribution, misurando il contributo di ogni parola/token all'output:

- Integrated Gradients (IG): Calcola la media dei gradienti lungo un percorso dal baseline all'input reale. È precisa per modelli differenziabili e mostra visivamente l'impatto dei token.
- LIME (Local Interpretable Model-agnostic Explanations): Modella la predizione originale con un modello interpretabile locale (es. regressione lineare) su versioni perturbate dell'input. È "model-agnostic", non richiede accesso ai gradienti del modello originale e stima l'influenza di ogni parola.
- SHAP (SHapley Additive exPlanations): Basato sulla teoria dei giochi, valuta il contributo di ogni parola (come un "giocatore") al risultato finale, considerando tutti i possibili sottoinsiemi di parole. Offre una

spiegazione equa e teoricamente fondata, ma è computazionalmente costosa.