Programmazione di Applicazioni Data Intensive
Linee guida progetti d'esame

Il progetto deve essere svolto individualmente oppure in gruppo fino a tre persone, con un impegno approssimativo di 2-3 giornate di lavoro a testa.
Il progetto deve riguardare l'**analisi di uno o più dataset** e l'addestramento da essi di un **modello di classificazione, regressione o recommendation**.
Il progetto va svolto e consegnato in forma di **file Jupyter** (.ipynb) contenente il codice e i risultati richiesti. Il file deve includere **commenti** sui dati analizzati, sulle scelte effettuate e sui risultati ottenuti; **non** vanno copiate o riportate nozioni già presenti nel materiale didattico. Per consegnare il progetto, inviare via e-mail ai docenti del corso **almeno 5 giorni prima della prova orale** un URL pubblicamente accessibile dove è possibile leggere il file Jupyter, utilizzando ad es. nbviewer, Colab, GitHub, ecc.

Passaggi minimi da svolgere

1. Descrivere in modo chiaro **il contesto e l'obiettivo del modello** di predizione, la fonte e la struttura del **dataset** utilizzato e le **variabili** che contiene. Eseguire eventualmente una prima scrematura dei dati, eliminando ad es. variabili palesemente non informative (es. identificatori) o con molti dati nulli.
2. Eseguire un'**analisi esplorativa** del dataset, riportando statistiche generali (medie, quartili, valori distinti, indici di correlazione, …) e distribuzioni delle variabili in tabelle e grafici (a torta, istogrammi, a dispersione, …). **Commentare adeguatamente** i risultati dell'analisi ed utilizzarli eventualmente per eliminare dati non utilizzabili nell'analisi.
3. **Preparare i dati** per l'addestramento e la validazione dei modelli di predizione: isolare la variabile "target" da predire e le variabili predittive, suddividere i dati in training e test set, eseguire eventuali operazioni di preprocessing come ad es. one-hot encoding di variabili categoriche e oversampling o undersampling in caso di classi sbilanciate.
4. **Addestrare e validare** due o più modelli di predizione, calcolandone le misure di performance viste nel corso (es. MSE, errore relativo e coefficiente R^2 per modelli di regressione) e analizzando il modello addestrato (es. coefficienti in una regressione lineare o nodi dei primi livelli di un albero decisionale) per individuare le variabili più o meno rilevanti nella predizione.
5. Scegliere uno o più modelli di base (es. regressione ridge) ed eseguire una **ricerca degli iperparametri** esaustiva (*grid search*) o a campione (*randomized search*) che massimizzi le performance del modello.

Dove possono essere reperiti i dataset?

- Da siti noti come ad es.
  - UCI Machine Learning Repository (https://archive.ics.uci.edu/ml)
  - Kaggle (https://www.kaggle.com/datasets)
  - OpenML (https://www.openml.org/search?type=data)
  - Registry of Open Data on AWS (https://registry.opendata.aws/)
- Ottenuti tramite l'utilizzo di API o librerie apposite
  - ad es. dataset di borsa accessibili con package come yfinance
- Tramite organizzazioni o aziende che ne abbiano autorizzato l'utilizzo

Il dataset scelto deve contenere una quantità adeguata di dati, indicativamente almeno qualche migliaio di istanze e una decina di variabili. Se il dataset è molto grande, è possibile selezionare un sottoinsieme casuale delle istanze per ridurre i tempi di calcolo.

Punti facoltativi

Per migliorare la valutazione del progetto, specialmente se svolto in gruppo, è possibile ad es.:

- utilizzare **molteplici dataset**, unendone insieme i dati in uno unico o utilizzandoli separatamente per addestrare diversi modelli;
- generare **nuove variabili** in aggiunta a quelle presenti nei dati, ad es. estraendo i singoli campi di una data (giorno della settimana, mese, …) o i termini chiave da campi testuali, si vedano l'esercitazione sulla predizione di borsa e quella col dataset Rossman per alcuni esempi;
- addestrare **più tipi di modelli** di predizione e confrontarne i risultati, avvalendosi eventualmente di librerie esterne come XGBoost, LightGBM, …
- eseguire una **validazione più approfondita** dei modelli, ad es. tramite la nested cross validation;
- creare una **applicazione Web** che consenta l'utilizzo dei modelli addestrati tramite interfaccia utente e/o API.