Progetti Machine Learning

A.A. 2024/25

1 Informazioni Generali

Tutti gli studenti che intendono sostenere l'esame di ML devono svolgere e discutere il progetto. La valutazione attribuita al progetto pesa per il 25% del voto finale. Il progetto può essere svolto singolarmente o in gruppi da 2 studenti.

Scelta della traccia. Di seguito sono descritte le tracce disponibili. Un numero limitato di studenti/gruppi può svolgere ciascuna traccia. Prima di iniziare a lavorare al progetto, è necessario richiedere l'assegnazione di una traccia, tramite il form che sarà pubblicato sul canale Teams, esprimendo le proprie preferenze. L'assegnazione terrà conto – per quanto possibile – delle preferenze espresse da ciascuno; ove necessario, le richieste saranno gestite con una logica *first-come-first-served*.

Discussione. La discussione consiste in una breve presentazione del lavoro svolto da parte degli studenti ed eventuali domande da parte dei docenti. La discussione del progetto deve avvenire:

- in data da concordare con i docenti, per chi ha superato le prove di esonero;
- per chi sostiene le prove scritte finali, contestualmente alla prova orale. Gruppi da 2 studenti dovrebbero preferibilmente svolgere l'orale lo stesso giorno. Nel caso non fosse possibile, il progetto sarà discusso da entrambi gli studenti contestualmente alla prima prova orale sostenuta.

Consegna. Gli studenti devono consegnare il codice della soluzione sviluppata e opportuna documentazione delle problematiche affrontate, scelte progettuali e valutazione delle prestazioni. Se la soluzione consiste in un notebook Jupyter, la documentazione del lavoro può essere inclusa nel notebook stesso. In alternativa, è possibile consegnare una breve relazione (2-3 pagine).

La consegna deve consistere di un **file singolo** (e.g., un notebook Jupyter, oppure un archivio ZIP contentente più file). Il nome del file dovrebbe contenere chiara indicazione dei cognomi degli studenti. La consegna avviene tramite **l'upload nella cartella Dropbox** seguente:
`https://www.dropbox.com/request/CdqHy0cBSRDsYSsb91sK`.

In caso di problemi, la consegna può avvenire via e-mail ai docenti. In questo caso, per evitare problemi con i filtri antivirus dei server di posta, anziché allegare il codice alla mail, si consiglia di caricare la soluzione su uno spazio di storage (es., Google Drive, Dropbox, OneDrive) e inserire nella mail solo un link al file.

Criteri di valutazione. I progetti saranno valutati rispetto a diversi obiettivi:

- chiarezza della documentazione e della presentazione;
- originalità e ragionevolezza delle soluzioni proposte;
- consapevolezza degli strumenti utilizzati;
- valutazione delle prestazioni dei modelli proposti (in termini di qualità delle predizioni e, se rilevante, costo computazionale);
- riusabilità della soluzione.

2 Tracce

Le tracce sono etichettate con le sigle `T1`, `T2`, etc. Di alcune tracce esistono varianti multiple, etichettate con le lettere `a`, `b`, etc.

Per alcune tracce, può essere opportuno utilizzare GPU per l'addestramento dei modelli. È possibile utilizzare una GPU (con alcuni limiti) gratuitamente tramite Google Colab. In questo caso, si consiglia di caricare il dataset di riferimento su Google Drive[1].

Dataset. I dati associati a ciascuna traccia – se non indicato diversamente – sono disponibili nel gruppo Teams del corso (canale "Lezioni", sezione Files, cartella "Dataset").

2.1 Traccia T1 - Sentiment Analysis: Recensioni Amazon

L'obiettivo del progetto è sviluppare un modello di ML per classificare recensioni di clienti Amazon, di cui viene fornito il testo originale in lingua inglese. In particolare, a seconda della traccia specifica, dovranno essere risolti 2 differenti problemi: (i) classificazione binaria delle recensioni *positive*/*negative*, (ii) classificazione delle recensioni come *positive*, *negative* o *neutrali*.

I dataset da utilizzare sono i seguenti:

- **T1a**: Class. binaria: `AmazonFashionBinary.zip`
- **T1b**: Class: multiclasse: `AmazonFashionMulti.zip`
- **T1c**: Class. binaria: `Grocery_and_Gourmet_FoodBinary.zip`

[1] `https://colab.research.google.com/notebooks/io.ipynb`

- **T1d**: Class: multiclasse: `Grocery_and_Gourmet_FoodMulti.zip`
- **T1e**: Class. binaria: `Tools_and_Home_ImprovementBinary.zip`
- **T1f**: Class: multiclasse: `Tools_and_Home_ImprovementMulti.zip`

Per ridurre il costo computazionale dell'addestramento, si suggerisce di limitare la lunghezza massima dei frammenti delle recensioni forniti in input al modello (con possibile impatto negativo sull'accuratezza).

Valutare l'accuratezza dei modelli su dati di validazione opportunamente estratti dal dataset, provando almeno 2 diverse configurazioni del modello (e.g., variando lunghezza dei frammenti di testo in input o la dimensione del vocabolario). Inoltre, verificare il funzionamento del modello su messaggi scritti da voi.

2.2 Traccia T2 - Sentiment Analysis: Twitter

L'obiettivo del progetto è sviluppare un modello di ML per classificare messaggi pubblicati sul social Twitter come *positivi* o *negativi*. Il dataset di riferimento, che contiene tweet in lingua inglese, è `TwitterParsed.zip`.

Valutare l'accuratezza dei modelli su dati di validazione opportunamente estratti dal dataset, provando almeno 2 diverse configurazioni del modello (e.g., variando lunghezza dei frammenti di testo in input, o la dimensione del vocabolario). Inoltre, verificare il funzionamento del modello su messaggi scritti da voi.

2.3 Traccia T3 - Riconoscimento efficiente di immagini di animali tramite transfer learning e quantizzazione

Il progetto si pone come obiettivo il riconoscimento di diverse specie di animali a partire da fotografie, utilizzando una rete neurale. Il dataset di riferimento (`animals.zip`) contiene immagini di animali precedentemente etichettate.

Il progetto richiede di sviluppare una soluzione con attenzione sia all'accuratezza del modello che al costo computazionale associato. Nello specifico, si richiede di:

- Utilizzare la tecnica del transfer learning per sfruttare un modello pre-addestrato (a scelta) come punto di partenza nello sviluppo della soluzione
- Studiare ed utilizzare, dopo l'addestramento, la tecnica della *quantizzazione dei pesi*, per ridurre la dimensione del modello e migliorare le prestazioni in fase di inferenza. A tal fine si raccomanda la lettura di:
  `https://ai.google.dev/edge/litert/models/post_training_integer_quant`
- Confrontare le prestazioni del modello (in termini sia di accuratezza che costo computazionale) con e senza quantizzazione

- **Solo T3a**: confrontare le prestazioni del modello nel caso le immagini di input siano in scala di grigi;
- **Solo T3b**: confrontare le prestazioni del modello ottenuto partendo da almeno 2 diversi modelli pre-addestrati.

2.4 Traccia T4 - Riconoscimento di animali da tracce audio

Si consideri un dataset contenente registrazioni audio. Si richiede di classificare le tracce riconoscendo la specie animale che l'ha prodotta. Il dataset fornito (`cats_dogs_dolphins.zip`) contiene suoni etichettati di 3 specie animali (i.e., cani, gatti e delfini).

Si richiede di sviluppare e confrontare 2 diversi modelli:

- un modello che prenda in input direttamente la forma d'onda associata a ciascuna traccia (i.e., sequenze)
- un modello che prenda in input lo *spettrogramma* di ciascuna traccia[2], ovvero la rappresentazione grafica dell'intensità di un suono in funzione del tempo e della frequenza. Per ottenere lo spettrogramma delle tracce audio si possono utilizzare numerose librerie esistenti, incluso Keras[3].

2.5 Traccia T5 - Diagnosi di polmonite

Data una raccolta (`chest_xray.zip`) di immagini etichettate relative a radiografie del torace, si richiede di sviluppare un modello di ML per la diagnosi automatizzata di casi di polmonite. Nello sviluppo del modello, si consiglia (1) l'utilizzo opportuno di tecniche di data augmentation[4] per ridurre l'overfitting, e (2) di ridimensionare preliminarmente le immagini in fase di costruzione del dataset, per ridurre il peso computazionale dell'addestramento.

Si richiede inoltre di:

- confrontare le prestazioni del modello in almeno 2 diverse configurazioni (a scelta), al variare delle tecniche di data augmentation utilizzate e/o della dimensione delle immagini in input;
- studiare una soluzione per permettere all'utente di esprimere un livello di preferenza verso la riduzione dei "falsi positivi" o dei "falsi negativi", senza la necessità di addestrare un nuovo modello.

[2] `https://it.wikipedia.org/wiki/Spettrogramma`
[3] `https://keras.io/api/layers/preprocessing_layers/audio_preprocessing/mel_spectrogram/`
[4] `https://keras.io/api/layers/preprocessing_layers/image_preprocessing/`

2.6 Traccia T6 - Explainable ML

C'è crescente interesse verso lo sviluppo di modelli di ML per le cui predizioni sia possibile derivare delle spiegazioni (e.g., il valore di quale feature ha determinato l'assegnazione ad una certa classe). La libreria `SHAP`[5], basata sul concetto del valore di Shapley, fornisce uno strumento semplice per l'intepretazione di molti modelli di ML.

Si richiede quindi di studiare il funzionamento della libreria SHAP, risolvere un task di classificazione o regressione su dati tabulari usando un modello a scelta, ed utilizzare SHAP per fornire una intepretazione delle predizioni ottenute su alcuni campioni estratti dal validation set (o dal test set, se disponibile).

Nello specifico, il task da risolvere è:

T6a Predizione della performance di studenti, sulla base del dataset (sintetico) `StudentPerformanceFactors.csv`. La colonna `Exam Score` rappresenta il target della predizione. Una descrizione delle feature è nel file `StudentPerformance.pdf`.

T6b Predizione del valore di automobili usate, sulla base del dataset `autoscout_prices.csv`. La colonna `price` rappresenta il target della predizione. Una descrizione delle feature è nel file `autoscout_prices.pdf`.

2.7 Traccia T7 - Intrusion Detection

Il dataset NSL-KDD (`KDDTrainClean.csv`) rappresenta un benchmark per sistemi di *Intrusion Detection*. Il dataset contiene informazioni sui flussi di traffico di rete verso una infrastruttura IT. Ciascun flusso è etichettato come "normale" o associato ad una tipologia di attacco. La colonna `label` del dataset rappresenta l'etichetta.

Si richiede di addestrare modelli di ML per i seguenti task:

T7a Classificazione binaria dei flussi (normale/attacco) utilizzando almeno 2 diversi modelli di ML. Classificare come "attacco" tutto ciò che non è etichettato come "normal" nel dataset.

T7b Classificazione dei flussi e il riconoscimento dei diversi attacchi (classificazione multi-classe).

T7c Riconoscimento di flussi "anomali" tramite l'utilizzo di un autoencoder addestrato sui dati non etichettati (i.e., ignorando la colonna `label`). Valutarne le prestazioni rispetto alle etichette presenti nel dataset.

[5] `https://shap.readthedocs.io/en/latest/index.html`

2.8 Traccia T8 - Compressione di immagini tramite Autoencoder

Addestrare un autoencoder per la compressione di immagini di fiori, utilizzando il dataset `102flowers.zip`. Confrontare le prestazioni del modello (e sue eventuali varianti) su un sottoinsieme di immagini di validazione, in termini di rapporto di compressione e qualità di ricostruzione dell'immagine. Per valutare la qualità delle immagini ricostruite, si suggerisce di utilizzare lo *Structural similarity index measure* (SSIM) (implementazioni per Tensorflow facilmente reperibili). Confrontare il modello anche con le immagini compresse utilizzando il formato JPEG con diverse configurazioni per il quality factor.

3 Suggerimenti: Caricamento dei Dati

I dataset indicati sono stati pre-processati per semplificarne il caricamento in Tensorflow e, in alcuni casi, per ridurne le dimensioni.

3.1 Immagini

Si consiglia di caricare i dataset usando la funzione `image_dataset_from_directory()`[6]. La funzione consente, tra le altre cose, anche di ridimensionare automaticamente le immagini in fase di caricamento. Sfruttare, se necessario, il ridimensionamento per (i) adattare la dimensione delle immagini ai requisiti di eventuali modelli pre-addestrati in uso, oppure (ii) ridurre il peso computazionale dell'addestramento.

3.2 Testo

Si consiglia di caricare i dataset usando la funzione `text_dataset_from_directory()`[7]. Le sequenze caricate possono essere tokenizzate usando il layer `TextVectorization` di Keras[8]. Di default, il layer utilizza un tokenizer a livello di parola. Il layer deve essere "adattato" sui dati di training prima di essere utilizzato (v. esempio seguente). Il layer consente anche di limitare la lunghezza massima delle sequenze prodotte e di configurare la dimensione del vocabolario utilizzato.

```
MAX_SEQUENCE_LENGTH = 50
vectorizationLayer = tf.keras.layers.TextVectorization(
    max_tokens=VOCAB_SIZE,
    output_mode='int',
    output_sequence_length=MAX_SEQUENCE_LENGTH)
```

[6] https://www.tensorflow.org/api_docs/python/tf/keras/preprocessing/image_dataset_from_directory

[7] https://www.tensorflow.org/api_docs/python/tf/keras/preprocessing/text_dataset_from_directory

[8] https://www.tensorflow.org/api_docs/python/tf/keras/layers/TextVectorization

```
# Make a text-only dataset (without labels), then call adapt
train_text = train_ds.map(lambda x, y: x)
vectorizationLayer.adapt(train_text)


model = tf.keras.models.Sequential()
model.add(vectorizationLayer)
```

3.3 Audio

Si consiglia di caricare i dataset usando la funzione `audio_dataset_from_directory()`[9]

[9] `https://www.tensorflow.org/api_docs/python/tf/keras/utils/audio_dataset_from_directory`