

# AI Developer Junior – Test Entry Level

## Progetto di Elaborazione Ordini tramite estrazione e corrispondenza

**Descrizione del progetto:** hai un'attività che si basa sulla vendita di prodotti attraverso WhatsApp business. Per semplicità, assumiamo che i messaggi siano sempre e solo ordini contenenti una lista di prodotti.

Di seguito degli esempi:

*5 panini, insalata mezzo chilo, mele 1 cassetta, <dialetto> 3 pezzi*

*5 panini  
insalata mezzo chilo  
mele 1 cassetta  
<dialetto> 3 pezzi*

### Compiti per il candidato:

1. **Analisi:** iniziare con un'analisi esplorativa descrivendo come approcceresti il problema.
2. **Prompt:** creare un prompt che analizzi un messaggio d'esempio ed estraiga le informazioni. L'output deve essere un dict con i seguenti campi: *name, quantity, unit*
3. **Ottimizzazione del risultato:** Confrontare il risultato ottenuto nello step precedente con quanto presente in database (alias compresi) al fine di ottenere la lista di prodotti esatta a cui fa riferimento il testo del messaggio: *id, nome, prezzo, unità, quantità*. Includi lo score dell'accuratezza del match.
4. **Presentazione dei risultati:** presentare i risultati del progetto e spiegare il processo seguito per sviluppare il modello.

### Strumenti a disposizione:

1. Python
2. OpenAI API Key
3. un database vettoriale (a tua scelta) contenente:
  - lista di prodotti (id [*intero*], name[*stringa*], price[*decimale*], unit [*stringa*]) (5, panino, 0.20, pezzo)
  - lista degli alias per i nomi dei prodotti (id [*intero*], aliases [*array di stringhe*])

### Condivisione:

Link repository pubblica (Git provider a scelta).

# AI Developer Junior – Test Medium Level [1]

## Progetto di Classificazione delle Immagini con TensorFlow e Keras

**Descrizione del progetto:** Hai a disposizione un dataset contenente immagini di animali domestici, tra cui cani e gatti. Il tuo obiettivo è sviluppare un modello di machine learning che possa classificare automaticamente se un'immagine contiene un cane o un gatto. Deve essere in grado di raggiungere un'accuratezza di almeno il 90% nella classificazione.

### Compiti per il candidato:

1. **Analisi dei dati:** iniziare con un'analisi esplorativa dei dati per comprendere la distribuzione delle classi, le dimensioni delle immagini, ecc.
2. **Preelaborazione dei dati:** preparare i dati per l'addestramento, inclusa la normalizzazione delle immagini, la suddivisione del dataset in set di addestramento e di test, e la creazione di generatori di dati.
3. **Sviluppo del modello:** progettare e allenare un modello di classificazione delle immagini utilizzando TensorFlow e Keras. Puoi sperimentare con diverse architetture di reti neurali convoluzionali (CNN) e ottimizzatori.
4. **Valutazione del modello:** valutare le prestazioni del modello utilizzando il set di test e misurare l'accuratezza, la precisione, il richiamo e l'F1-score.
5. **Ottimizzazione del modello:** Se il modello iniziale non raggiunge il 90% di accuratezza, cercare di ottimizzarlo. Puoi eseguire l'ottimizzazione dei parametri, provare tecniche come l'augmentation dei dati, regolarizzazione, dropout, ecc.
6. **Presentazione dei risultati:** presentare i risultati del progetto e spiegare il processo seguito per sviluppare il modello.

### DATASET:

- <https://www.microsoft.com/en-us/download/details.aspx?id=54765>
- oppure
- <https://storage.googleapis.com/mledu->

### Condivisione:

Link repository pubblica (Git provider a scelta).

# AI Developer Junior – Test Medium Level [2]

## Progetto di Elaborazione del Linguaggio Naturale (NLP) con Python e Scikit-Learn

**Descrizione del progetto:** Hai a disposizione un dataset contenente recensioni di film e le relative etichette di sentiment, ovvero se una recensione è positiva o negativa. Il tuo obiettivo è sviluppare un modello di machine learning per la classificazione dei sentiment delle recensioni utilizzando tecniche di NLP. Deve essere in grado di raggiungere un'accuratezza di almeno l'85% nella classificazione dei sentiment.

### Compiti per il candidato:

- **Esplorazione dei dati:** iniziare con un'analisi esplorativa dei dati per comprendere la distribuzione delle etichette di sentiment, la lunghezza delle recensioni, ecc.
- **Preelaborazione dei dati:** eseguire la preelaborazione dei dati, inclusa la rimozione di stopwords, la tokenizzazione delle frasi, la creazione di vettori di parole (word embeddings) e la suddivisione del dataset in set di addestramento e di test.
- **Sviluppo del modello:** progettare e allenare un modello di machine learning per la classificazione dei sentiment utilizzando Scikit-Learn. Puoi sperimentare con algoritmi come Support Vector Machines (SVM), Naive Bayes, o modelli di deep learning come reti neurali ricorrenti (RNN) o Long Short-Term Memory (LSTM) se si sentono confortevoli.
- **Valutazione del modello:** valutare le prestazioni del modello utilizzando il set di test e misurare l'accuratezza, la precisione, il richiamo e l'F1-score.
- **Ottimizzazione del modello:** Se il modello iniziale non raggiunge l'85% di accuratezza, cercare di ottimizzarlo. Puoi eseguire l'ottimizzazione dei parametri, provare diverse configurazioni di algoritmi o esplorare l'uso di word embeddings pre-addestrati.
- **Presentazione dei risultati:** presentare i risultati del progetto e spiegare il processo seguito per sviluppare il modello.

### DATASET:

<https://ai.stanford.edu/%7Eamaas/data/sentiment/>

### Condivisione:

Link repository pubblica (Git provider a scelta).