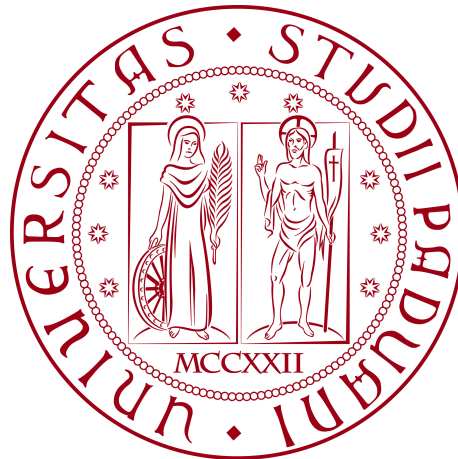


Università degli Studi di Padova

DIPARTIMENTO DI MATEMATICA “TULLIO LEVI-CIVITA”

CORSO DI LAUREA IN INFORMATICA



## Test Automatici con Large Language Model

*Tesi di Laurea Triennale*

*Relatore*

Prof. Ballan Lamberto

*Laureando*

Dugo Alberto

Matricola 2042382

---

ANNO ACCADEMICO 2023-2024



# Ringraziamenti

Padova, Luglio 2024

*Dugo Alberto*

# Abstract

Il presente documento illustra l'attività di stage svolta dal laureando Alberto Dugo presso l'azienda Zucchetti Spa.

Durante il periodo di stage, della durata di 320 ore, vi è stata l'opportunità di approfondire le conoscenze in ambito **machine learning**<sub>G</sub>. In particolare era richiesto lo studio e l'implementazione di test automatici derivanti direttamente dal codice e dalla documentazione, sfruttando le abilità dei sistemi di intelligenza artificiale ed in particolare del **Large Language Model (LLM)**<sub>G</sub>. Vi è stato inoltre la possibilità di fare **fine-tuning**<sub>G</sub> dei modelli **LLM** attraverso il metodo **Low-Rank Adaptation (LoRA)**<sub>G</sub> e quantizzare i modelli stessi in modo da renderli più efficienti.



# Indice

<b>1</b>	<b>Introduzione</b>	<b>1</b>
1.1	L'azienda . . . . .	1
1.2	Il progetto . . . . .	1
1.3	Organizzazione del testo . . . . .	2
<b>2</b>	<b>Processi e metodologie</b>	<b>4</b>
2.1	Processo di sviluppo del prodotto . . . . .	4
2.2	Strumenti utilizzati . . . . .	4
<b>3</b>	<b>Svolgimento del progetto</b>	<b>5</b>
3.1	Analisi del dominio applicativo . . . . .	5
3.1.1	Analisi del tema . . . . .	5
3.1.2	Esempi di utilizzo . . . . .	5
3.1.3	<i>Assured LLMSE</i> . . . . .	5
3.2	Analisi dei requisiti . . . . .	5
3.2.1	Analisi preventiva dei rischi . . . . .	5
3.2.2	Requisiti e obiettivi . . . . .	6
3.3	Sviluppo del prodotto . . . . .	7
3.3.1	<i>Script</i> . . . . .	7
3.3.2	<i>Benchmarks</i> . . . . .	7
3.3.3	<i>Fine-tuning</i> . . . . .	7
3.3.4	Quantizzazione . . . . .	7
3.3.5	Documentazione e test automatici . . . . .	7
3.4	Resoconto finale . . . . .	7

## INDICE

---

3.4.1	Prodotti ottenuti . . . . .	7
3.4.2	Risultati ottenuti . . . . .	7
3.4.3	Conclusione . . . . .	7
<b>4</b>	<b>Valutazione retrospettiva</b>	<b>8</b>
4.1	Conoscenze acquisite . . . . .	8
4.2	Valutazione personale . . . . .	8
	<b>Bibliografia</b>	<b>i</b>
	<b>Sitografia</b>	<b>ii</b>
	<b>Acronimi e abbreviazioni</b>	<b>iii</b>
	<b>Glossario</b>	<b>iv</b>

## Elenco delle figure

1.1	logo di Zucchetti . . . . .	1
-----	-----------------------------	---

## Elenco delle tabelle

3.1	Requisiti. . . . .	6
-----	--------------------	---



# Capitolo 1

## Introduzione

### 1.1 L'azienda



**Figura 1.1:** logo di Zucchetti

Zucchetti S.p.a. è la prima software house in Italia per fatturato, opera nel settore dell'*Information Technology*, ed è stata fondata nel 1978 da Fabrizio Bernini a Lodi. L'azienda è specializzata nella realizzazione di software gestionali per pianificazione delle risorse d'impresa, soluzioni per il controllo degli accessi e sistemi di automazione industriale. Al giorno d'oggi Zucchetti oltre ad avere sedi in tutta Italia è presente anche in 50 paesi esteri, tra cui Cina, Germania, USA e Svizzera e conta più di 8.000 dipendenti e più di 1650 *partner*.

### 1.2 Il progetto

Lo *stage* si svolgerà presso l'azienda Zucchetti, con sede a Padova. Il progetto di *stage* prevede la ricerca e lo sviluppo di test automatici derivanti direttamente dal codice e dalla documentazione, sfruttando le abilità dei sistemi di *intelligenza artificiale* ed in particolare dei **Large Language Models**. Lo *stage* sarà diviso

in due macroperiodi di quattro settimane ciascuno. Nel primo periodo lo studente dovrà analizzare attraverso lo studio di paper accademici e documentazioni le tecniche di *testing* che sfruttano **Large Language Models**. Successivamente dovrà implementare un **prototipo** di generatore di test automatici in *Python* che utilizza un modello di *intelligenza artificiale*. Nella seconda parte del primo periodo lo studente dovrà decorare il codice attraverso commenti e generare *test*. I risultati di questi verranno poi confrontati con quelli ottenuti dalla prima parte di periodo. Nel secondo periodo invece lo studente dovrà implementare un sistema che, a partire da pseudo codice riesca a generare *test* automatici. Dovrà inoltre fare **fine-tuning** dei modelli **LLMs** con il metodo **LoRA**, per poi confrontare i risultati ottenuti con quelli nel primo periodo.

### 1.3 Organizzazione del testo

**Il secondo capitolo** descrive i processi e le metodologie utilizzate durante lo *stage*. In particolare si approfondiranno i processi di sviluppo *software* e gli strumenti utilizzati per fare ciò.

**Il terzo capitolo** si propone di delineare il dominio applicativo del progetto, mediante un'analisi dettagliata del tema accompagnata da esempi pratici di utilizzo. Inoltre, si provvederà a fornire una descrizione esaustiva del funzionamento di *Assured LLMs*. In questa sezione, sarà altresì redatta una lista esaustiva di rischi, requisiti e obiettivi del progetto. Successivamente, si procederà con la descrizione del prodotto sviluppato, che includerà *script*, *benchmarks*, **fine-tuning**, quantizzazione e documentazione. Infine, si condurrà una valutazione conclusiva dei risultati ottenuti.

**Il quarto capitolo** concluderà il documento, presentando una valutazione retrospettiva personale dell'esperienza di *stage* e delle conoscenze acquisite.

In seguito si possono trovare le convenzioni tipografiche utilizzate per la stesura del documento:

- gli acronimi, le abbreviazioni e i termini ambigui o di uso non comune menzionati vengono definiti nel glossario, situato alla fine del presente documento;
- per la prima occorrenza dei termini riportati nel glossario viene utilizzata la seguente nomenclatura: *parola* $\mathbf{G}$ ;
- i termini in lingua straniera o facenti parti del gergo tecnico sono evidenziati con il carattere *corsivo*.

# Capitolo 2

## Processi e metodologie

### 2.1 Processo di sviluppo del prodotto

### 2.2 Strumenti utilizzati

gli strumenti utilizzati per la realizzazione del progetto sono stati:

- **Git**: sistema di controllo di versione distribuito;
- **GitHub**: servizio di hosting per progetti software che utilizzano Git;
- **Google Colab**: servizio di Google che permette di eseguire codice Python in cloud;
- **Hugging Face**: libreria Python che fornisce modelli di Machine Learning pre-addestrati;
- **PyCharm**: **Integrated Development Environment (IDE)**<sub>G</sub> per lo sviluppo in Python;
- **Python**: linguaggio di programmazione ad alto livello, interpretato, interattivo, orientato agli oggetti, adatto per lo sviluppo di applicazioni legate al machine learning;

# Capitolo 3

## Svolgimento del progetto

### 3.1 Analisi del dominio applicativo

#### 3.1.1 Analisi del tema

Il tema del progetto riguarda la realizzazione di test automatici per il testing di codice sorgente, ponendosi come obiettivo la semplificazione del processo di testing affidato ai programmatori attraverso l'utilizzo di **Large Language Models**.  
descrive il dominio del problema da affrontare: la porzione del mondo reale, rilevante per il sistema

-> Su cui si devono mantenere informazioni

-> Concuissideve interagire

#### 3.1.2 Esempi di utilizzo

#### 3.1.3 *Assured LLMSE*

### 3.2 Analisi dei requisiti

#### 3.2.1 Analisi preventiva dei rischi

Durante la fase di analisi dei rischi sono stati individuate le possibili criticità che potranno essere riscontrate. Si è quindi proceduto a elaborare delle possibili soluzioni per far fronte a tali rischi.

**1. Mancanza di materiale informativo**

**Descrizione:** Trattandosi di una novità nel settore e in fase di crescita, la possibile assenza di materiale informativo relativo all'argomento stesso potrebbe rallentare il processo di apprendimento..

**Soluzione:** coinvolgimento del responsabile a capo del progetto relativo.

**3.2.2 Requisiti e obiettivi**

Obiettivo	Descrizione
OB 1	Realizzazione di smoke test in Python generati da codice reale.
OB 2	Realizzazione di decorazioni assert per funzioni.
OB 3	Realizzazione di test a partire da pseudocodice.
OB 4	Test con modello LLM con fine tuning attraverso LoRA.

**Tabella 3.1:** Requisiti.

## 3.3 Sviluppo del prodotto

### 3.3.1 *Script*

### 3.3.2 *Benchmarks*

### 3.3.3 *Fine-tuning*

### 3.3.4 Quantizzazione

### 3.3.5 Documentazione e test automatici

## 3.4 Resoconto finale

### 3.4.1 Prodotti ottenuti

### 3.4.2 Risultati ottenuti

### 3.4.3 Conclusione

## Capitolo 4

### Valutazione retrospettiva

#### 4.1 Conoscenze acquisite

#### 4.2 Valutazione personale



# Bibliografia

## Testi

James P. Womack, Daniel T. Jones. *Lean Thinking, Second Editon*. Simon & Schuster, Inc., 2010.

## Articoli

Einstein, Albert, Boris Podolsky e Nathan Rosen. «Can Quantum-Mechanical Description of Physical Reality be Considered Complete?» In: *Physical Review* 47.10 (1935), pp. 777–780. DOI: [10.1103/PhysRev.47.777](https://doi.org/10.1103/PhysRev.47.777).

# Sitografia

*Manifesto Agile.* URL: <http://agilemanifesto.org/iso/it/>.

# Acronimi e abbreviazioni

**IDE** Integrated Development Environment. i, iv, 4

**LLM** Large Language Model. i, iv, 2

**LoRA** Low-Rank Adaptation. i, iv, 2

# Glossario

**Fine-tuning** ADD DESCRIPTION.. [i](#), [iv](#), [2](#)

**IDE** Un Integrated Development Environment (IDE) è un'applicazione software che fornisce servizi per facilitare lo sviluppo di software. Un IDE generalmente comprende un editor di codice sorgente, strumenti di compilazione e debugging e un ambiente per eseguire il software in sviluppo.. [i](#), [iii](#)

**LLM** Un Large Language Model (LLM) è un modello capace di generare testi in linguaggio naturale basandosi su modelli statistici. Questi modelli acquisiscono una conoscenza linguistica attraverso l'apprendimento di relazioni statistiche durante un processo di addestramento computazionalmente oneroso.. [i](#), [iii](#), [1](#), [2](#), [5](#)

**LoRA** ADD DESCRIPTION.. [i](#), [iii](#)

**Machine learning** branca dell'*intelligenza artificiale* che utilizza metodi statistici per migliorare la performance di un algoritmo nell'identificare pattern nei dati, imparando da questi a svolgere delle funzioni piuttosto che attraverso la programmazione esplicita.. [i](#), [iv](#)

**Prototipo** Un prototipo è un esemplare o un modello di un prodotto o di un sistema che viene realizzato antecedentemente al prodotto finale. [i](#), [2](#)