

# Giorgio De Simone

## Curriculum Vitæ

Via San Rocco 29

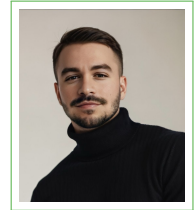
64028

Silvi (TE)

+39 3343094980

+39 0859353081

giogron28@gmail.com



### Personal information

Date of birth **28/05/1993.**

Nationality **Italian.**

### Experience

#### Vocational

September 2021 to Today **AI Engineer, Open-Source AI Developer, Italy, Remote Working.**

ROLE: Open-Source AI Developer.

MAIN RESPONSIBILITIES: Open Source AI repositories/models/manuals author and developer.

MAIN ACTIVITIES: Developing open-source AI repositories on GitHub ([link](#)) in the fields of Search Strategies (FundamentalsOfAI [link](#)), Swarm Intelligence (NetLogo [link](#)), Planning (Graphplan [link](#) and BlackBox [link](#)), First Order Logic (Prolog [link](#)), Machine Learning (Simple\_Linear\_Regression [link](#), Multiple\_Linear\_Regression [link](#), Logistic\_Regression [link](#) and Multi-class\_Classification [link](#)). Developing open-source AI models on Hugging Face ([link](#)) in the field of Machine Learning (Simple\_Linear\_Regression [link](#), Multiple\_Linear\_Regression [link](#), Logistic\_Regression [link](#) and Multi-class\_Classification [link](#)). Authoring and releasing open-source AI manuals on GitHub ([link](#)) in the fields of Delay Differential Equations (Delay numerical modeling of epidemiology [link](#)), Python programming (Introduction to Algorithms and Programming [link](#)), Reinforcement Learning (Cognition and Neuroscience [link](#)), Search Strategies, Swarm Intelligence, Planning and Constraints (Fundamentals of Artificial Intelligence and Knowledge Representation [link](#)).

TOOLS & TECHNOLOGIES:

- Programming languages
  - Python & C & C++ & Java;
- Logic programming language:
  - Prolog;
- Agent-based modeling programming language & Agent-based modeling IDE
  - NetLogo;
- Machine Learning libraries
  - Scikit-learn;
  - Pandas;
  - NumPy;
  - Matplotlib;
- Automated Planning Algorithms
  - Graphplan & BlackBox & Fast Forward;
- Automated Planning language:
  - PDDL;
- Integrated Development Environments
  - PyCharm & Visual Studio Code;
- Containerization platforms
  - Docker;
- Version control systems & CI/CD Execution
  - Git & GitHub Actions;
- Developer platform
  - GitHub & Hugging Face.

June 2024 to **QA Engineer, VIVATICKET**, Bologna, Italy, Remote Working.

Today ROLE: Employee at Vivaticket.

MAIN RESPONSIBILITIES: Ticketing solutions and Services – Technology Solutions.

MAIN ACTIVITIES: Designing, developing, and implementing automated testing solutions to ensure the quality and reliability of software applications, focusing on automating repetitive and manual testing tasks, allowing for faster and more efficient testing processes; specifically conducting Requirement Analysis, Tool and Framework Selection, Test Automation Planning, Scripting and Coding, Test Data Management, Execution and Reporting, Regression Testing, E2E Testing, Maintenance and Updates, CI/CD Execution, Documentation.

EMPLOYED AS: Full-time employee; permanent employment contract.

TOOLS & TECHNOLOGIES:

- Cloud computing services
  - Google Cloud Platform (GCP);
- Programming languages
  - Python;
- Integrated Development Environments
  - PyCharm;
- Frameworks
  - Robot Framework;
  - Browser;
  - Playwright;
- Behavior-driven developments
  - User story;
  - Gherkin;
  - Behave;
- Test reporting and analytics platforms
  - Report Portal;
- Containerization platforms
  - Docker;
- Container orchestration systems
  - Kubernetes;
- Version control systems
  - Git;
- Developer platform
  - GitHub;
- CI/CD Execution
  - GitHub Actions;
- Issue tracking systems
  - Jira;
- Collaborative software
  - Confluence.

January 2023 to May 2024 **QE Performance Engineer**, *EMM SYSTEM CONSULTING*, Naples, Italy, Remote Working.

ROLE: Consultant at Walgreens Boots Alliance.

MAIN RESPONSIBILITIES: Pharmacy and Healthcare Product Engineering – Technology Solutions.

MAIN ACTIVITIES: Evaluating and ensuring the performance, scalability, and reliability of software applications; specifically conducting Performance Testing, Benchmarking, Identifying Bottlenecks, Scalability Testing, Performance Monitoring, Automation, Reporting and Documentation.

EMPLOYED AS: Full-time employee; permanent employment contract.

TOOLS & TECHNOLOGIES:

- Cloud computing services
  - Microsoft Azure;
- Programming languages
  - Python;
  - Groovy;
- Integrated Development Environments
  - PyCharm;
  - Android Studio;
- Application Programming Interfaces
  - REST API;
- Collaboration platforms for API development
  - Postman;
- Load Testing Tools
  - Apache JMeter;
- Database systems
  - NoSQL;
- Domain-specific languages
  - SQL;
- Container orchestration systems
  - Kubernetes;
- Version control systems
  - Git;
- Developer platform
  - Azure DevOps;
- Issue tracking systems
  - Jira;
- Collaborative software
  - Confluence.

July 2022 to January 2023 **RPA Engineer**, *FORFIRM*, Manno, Switzerland, On-site.

ROLE: Consultant at Cornèr Bank.

MAIN RESPONSIBILITIES: Banking Software Products – Technology Solutions.

MAIN ACTIVITIES: Designing, developing, implementing, and maintaining Robotic Process Automation solutions. Developing software robots (bots) to automate repetitive, rule-based tasks, typically performed by humans; specifically conducting Process Analysis, RPA Solution Design, Development of RPA Bots, Integration, Implementation and Deployment, Maintenance and Optimization, Security and Compliance, Training and Support.

EMPLOYED AS: Full-time employee; permanent employment contract.

TOOLS & TECHNOLOGIES:

- Programming languages
  - AutoHotkey;
  - JavaScript;
- Integrated Development Environments
  - AHK Studio;
- Document Processing Platform
  - Hyperscience;
- Version control systems
  - Git.

January 2022 **QA Performance Engineer**, *EMM SYSTEM CONSULTING*, Naples, Italy, Remote Working.  
to July 2022

ROLE: Consultant at Walgreens Boots Alliance.

MAIN RESPONSIBILITIES: Pharmacy and Healthcare Product Engineering – Technology Solutions.

MAIN ACTIVITIES: Evaluating and ensuring the performance and reliability of software applications; specifically conducting Performance Testing, Automation, Reporting and Documentation.

EMPLOYED AS: Full-time employee; permanent employment contract.

TOOLS & TECHNOLOGIES:

- Cloud computing services
  - Microsoft Azure;
- Programming languages
  - Groovy;
- Integrated Development Environments
  - Visual Studio Code;
- Application Programming Interfaces
  - REST API;
- Load Testing Tools
  - Apache JMeter;
- Database systems
  - NoSQL;
- Domain-specific languages
  - SQL;
- Container orchestration systems
  - Kubernetes;
- Version control systems
  - Git;
- Issue tracking systems
  - Jira;
- Collaborative software
  - Confluence.

September **Test Performance Engineer**, *ALTEN*, Bologna, Italy, Remote Working.  
2021 to

ROLE: Consultant at Mediolanum Bank.

December MAIN RESPONSIBILITIES: Banking Software Products – Technology Solutions.

2021 MAIN ACTIVITIES: Evaluating and ensuring the performance of software applications; specifically conducting Performance Testing, Automation, Reporting and Documentation.

EMPLOYED AS: Apprentice.

TOOLS & TECHNOLOGIES:

- Software Testing Tools
  - LoadRunner – VuGen;
  - LoadRunner – Controller;
  - LoadRunner – Analysis;
- Programming languages
  - C;
  - C++;
- Application Programming Interfaces
  - REST API;
- Collaboration platforms for API development
  - Postman;
- Version control systems
  - Git.

May 2021 to August 2021 **Test Automation Engineer**, *ALTEN*, Milan, Italy, Remote Working.  
 ROLE: Consultant at Mediolanum Bank.  
 MAIN RESPONSIBILITIES: Banking Software Products – Technology Solutions.  
 MAIN ACTIVITIES: Designing, developing, and implementing automated testing solutions to ensure the quality and reliability of software applications, focusing on automating repetitive and manual testing tasks, allowing for faster and more efficient testing processes; specifically conducting Requirement Analysis, Tool and Framework Selection, Test Automation Planning, Scripting and Coding, Test Data Management, Execution and Reporting, Regression Testing, Maintenance and Updates, Documentation.  
 EMPLOYED AS: Trainee.  
 TOOLS & TECHNOLOGIES:

- Frameworks
  - Selenium;
  - Appium;
  - Katalon;
- Programming languages
  - Java;
  - Groovy;
- Behavior-driven developments
  - Cucumber;
  - Gherkin;
- Version control systems
  - Git;
- Integrated Development Environments
  - IntelliJ Idea;
  - Android Studio;
  - Eclipse.

## Education

September 2021 **Admission to second cycle degree in Artificial Intelligence**, *University of Bologna*, Bologna, Italy.  
*Skills: Search Strategies, Swarm Intelligence (NetLogo), Planning (Graphplan and BlackBox), First Order Logic (Prolog), Cognition, Neuroscience, Reinforcement Learning, Ability To Write An Algorithm, Computer Programming, Functional Programming, Procedural Programming, Python (Programming Language), Object-Oriented Programming (OOP).*  
 Click [here](#) to the certificate section on GitHub.

March 2021 **Degree in Mathematics**, *University of L'Aquila*, L'Aquila, Italy.  
*Grade: 108/110.*  
*Skills: Mathematical Analysis, Calculus, Differential Equations, Linear Algebra, Algebra, Probability, Numerical Analysis, Matlab, C (Programming Language), C++, Java, Mechanics and Thermodynamics, Electromagnetism, Differential Geometry, Topology, Cryptography, Magma (Computational Algebra System), Algebraic Topology, Quantum Mechanics, Functional Analysis, LaTeX.*  
 Click [here](#) to the certificate section on GitHub.

August 2014 **IELTS**, *British Council – IDP – Cambridge Assessment English*, Cork, Ireland.  
*Grade: 6.0.*  
*Skills: English language.*  
 Click [here](#) to the certificate section on GitHub.

July 2014 **Certificate of Achievement of IELTS Preparation Course**, *Cork English College*, Cork, Ireland.  
*Level: B2+ Upper Intermediate.*  
*Skills: English language.*  
 Click [here](#) to the certificate section on GitHub.

2013 to 2014 **Full time student of English**, *Rosscarbery English School*, Rosscarbery, Ireland.  
 Certificates: **Letter of reference.**  
 Click [here](#) to the certificate section on GitHub.

July 2012 **High School Diploma in Classical and Ancient Studies**, *Liceo Classico Gabriele D'Annunzio*, Pescara, Italy.  
Grade: **95/100**  
Click [here](#) to the certificate section on GitHub.

## Thesis

Title ***Delay numerical modeling of epidemiology.***

Supervisors Raffaele D'Ambrosio, Full Professor in Numerical Analysis, DISIM, University of L'Aquila.

Description The thesis concerns the SARS-CoV2 (COVID-19) pandemic that, starting from the end of February 2020, began spreading along the Italian peninsula. The main purpose of this thesis is the creation of a forecast model — based on Delay differential equations (DDEs) — that manages to alert the decision-making bodies and, in particular, the healthcare system, to hinder the emergence of any other pandemic outbreaks, or the arrival of subsequent pandemic waves. A new mathematical model to describe the pandemic is given. The model includes the class of undiagnosed infected people, and has a multi-region extension, to cope with the in-time and in-space heterogeneity of the epidemic. As a result we obtain a robust and reliable tool for the forecast of the total and active cases, which can be also used to simulate different scenarios. Lastly, we are able to address a number of issues, such as assessing the adoption of the lockdown in Italy, started from 11 March 2020, and how to employ a rapid screening test campaign for containing the epidemic.

Released March 2021

Link Click [here](#) to the document section on GitHub.

## Scientific writings

Title ***Fundamentals of Artificial Intelligence and Knowledge Representation.***

Description Notes, written by me, related to the 'Fundamentals of Artificial Intelligence and Knowledge Representation — Module 1' course (Second Cycle Degree/Two-Year Master in Artificial Intelligence) – offered by the University of Bologna and held by Professor Michela Milano – which concerns rudiments of artificial intelligence and knowledge representation.

Released December 2025

Link Click [here](#) to the document section on GitHub.

Associated Repositories FundamentalsOfAI ([link](#)), NetLogo ([link](#)), Graphplan ([link](#)), BlackBox ([link](#)), Prolog ([link](#)).

Title ***Cognition and Neuroscience.***

Description Notes, written by me, related to the 'Cognition and Neuroscience' course (Second Cycle Degree/Two-Year Master in Artificial Intelligence) – offered by the University of Bologna and held by Professor Francesca Starita and Professor Giuseppe Di Pellegrino – which concerns rudiments of cognition and neuroscience.

Released January 2024

Link Click [here](#) to the document section on GitHub.

Title	<b>Introduction to Algorithms and Programming.</b>
Description	Notes, written by me, related to the 'Introduction to Algorithms and Programming' course (Second Cycle Degree/Two-Year Master in Artificial Intelligence) – offered by the University of Bologna and held by Professor Simone Martini – which concerns rudiments of programming in Python.
Released	June 2023
Link	Click <a href="#">here</a> to the document section on GitHub.

## Licenses & certifications

- July 2025 **GitHub Actions, GitHub & Microsoft Learn & Pearson Vue.**  
*Skills: Continuous Integration (CI), Continuous Delivery (CD), Workflow Automation, Software Development Automation, Git, GitHub, GitHub Actions, Build Pipeline, Build Automation, Test Automation, Deployment Automation, Event-Driven Automation, Custom Workflow Creation, DevOps, Labeling Automation, Cross-Platform Workflow Execution, Self-Hosted Runner Configuration, Cloud Infrastructure Integration, Scalability in Automation, Security and Permissions in Automated Workflows.*  
*Description: The 'GitHub Actions' certification validates your proficiency in automating end-to-end software development workflows using GitHub's integrated CI/CD platform. It covers building, testing, and deploying applications directly from GitHub, creating flexible and event-driven workflows, and streamlining collaboration. You'll learn to automate tasks like labeling issues, integrate DevOps practices, and run workflows on GitHub-hosted or self-hosted runners across multiple operating systems.*  
Click [here](#) to the certificate section on GitHub.  
Click [here](#) to the certificate section on Microsoft Learn.  
Credential ID: 297B21-49JE0E
- December 2024 **GitHub Foundations, GitHub & Credly.**  
*Skills: Build Pipeline, Continuous Delivery (CD), Continuous Integration (CI), DevOps, Git, GitHub, GitHub Actions, Release Management, Collaboration Features, Modern Development, Project Management, Privacy, Security, Administration.*  
*Description: The 'GitHub Foundations' is a certification designed to introduce you to the fundamental concepts and products of GitHub. You'll discover the benefits of using GitHub as a collaborative platform and explore its core features, such as repository management, commits, branches, and merging. You'll gain a solid understanding of GitHub's essential tools and be well-equipped to start contributing to projects and collaborating effectively within GitHub.*  
Click [here](#) to the certificate section on GitHub.  
Click [here](#) to the certificate section on Credly.  
Credential ID: 0NKRMkyx
- July 2024 **[AZ-900] Microsoft Azure Fundamentals, Microsoft Learn & Pearson Vue.**  
*Skills: Describe cloud concepts, Describe Azure architecture and services, Describe Azure management and governance.*  
*Description: As a candidate for this certification, you're a technology professional who wants to demonstrate foundational knowledge of cloud concepts in general and Microsoft Azure in particular. This certification is a common starting point in a journey towards a career in Azure. You can describe Azure architectural components and Azure services, such as: Compute, Networking, Storage. You can also describe features and tools to secure, govern, and administer Azure. You should have skills and experience working with an area of IT, such as: Infrastructure management, Database management, Software development.*  
Click [here](#) to the certificate section on GitHub.  
Click [here](#) to the certificate section on Microsoft Learn.  
Credential ID: 42D8197E9B2001F2

- January 2024 **[PCEP-30-02] PCEP – Certified Entry-Level Python Programmer**, OpenEDG Python Institute.  
*Skills:* Ability To Write An Algorithm, Computer Programming, Functional Programming, Procedural Programming, Python (Programming Language).  
*Description:* Earners of the PCEP – Certified Entry-Level Python Programmer certification have knowledge of the universal concepts of computer programming, the syntax and semantics of the Python language as well as demonstrate the ability to accomplish coding tasks related to the essentials of programming in the Python language and fundamental skills in resolving typical implementation challenges with the help of the Python Standard Library.  
Click [here](#) to the certificate section on GitHub.  
Click [here](#) to the certificate section on OpenEDG Python Institute.  
Credential ID: Vj97.sc36.h2B1
- August 2014 **IELTS**, British Council – IDP – Cambridge Assessment English, Cork, Ireland.  
*Grade:* **6.0**.  
*Skills:* English language.  
Click [here](#) to the certificate section on GitHub.  
Credential ID: 14IE000544DEG899A

---

## Projects

- December 2025 **Multi-class Classification: From Theory to Interactive Web App**, GitHub & Hugging Face.  
*Skills:* Machine Learning, Scikit-Learn, Pandas, NumPy, Matplotlib  
*Description:* This project is a toolkit for solving multi-class classification problems, delivered through two key components: an educational Jupyter Notebook and a production-ready Streamlit web application. The web app allows users to perform complex classification analysis on their own CSV data with no coding required. Unlike standard binary classifiers, this tool is designed to handle targets with three or more categories (e.g., medical grades, product types, or risk levels). It features a robust workflow: data upload, automatic feature type detection, and advanced preprocessing—including missing value imputation, standard scaling for numerical features, and one-hot encoding for categorical variables. Users can train models using specific heuristic strategies like One-vs-Rest (OvR) and One-vs-One (OvO). The Jupyter Notebook offers a guide to the theory and practice behind the app, using the provided Obesity Risk Prediction dataset as a case study. It explores the mathematical challenges of adapting binary Logistic Regression for multi-class scenarios. The notebook details the specific strategies used to decompose multi-class problems: One-vs-Rest (OvR) — training a single classifier per class against all others — and One-vs-One (OvO) — training classifiers for every unique pair of classes. The notebook covers the full ML lifecycle using Python, scikit-learn, and Pandas, illustrating how proper scaling and encoding are critical for model convergence in multi-dimensional spaces.  
Click [here](#) to the Live Web App (Hugging Face space).  
Click [here](#) to the GitHub Repository (Code & Notebook).  
Click [here](#) for a comprehensive deep-dive into the theory and implementation details in my full article titled 'Beyond Yes or No: Mastering Multi-class Classification with OvR and OvO Strategies'.



October 2025 **Logistic Regression: From Theory to Interactive Web App, GitHub & Hugging Face.**

*Skills: Machine Learning, Scikit-Learn, Pandas, NumPy, Matplotlib*

*Description: This project is a toolkit for binary classification using logistic regression, delivered through two key components: an educational Jupyter Notebook and a production-ready Streamlit web application. The web app, deployed on Hugging Face, allows users to perform classification analysis on their own CSV data with no coding required. It features a full workflow: data upload, selection of independent variables, model training, and in-depth evaluation. The evaluation provides key classification metrics, including an interactive Confusion Matrix, a full Classification Report (Precision, Recall, F1-Score), Accuracy, Log Loss, and a Feature Coefficients plot to interpret model decisions. The Jupyter Notebook offers a guide to the theory and practice behind the app, using the provided ChurnData.csv as an example. It explains how logistic regression models the probability of a binary outcome (e.g., 1 for 'Churn' or 0 for 'Stay') based on one or more independent variables. The goal is to find the optimal sigmoid function (the "S-curve") that maps a linear combination of inputs to a probability between 0 and 1. This is achieved by minimizing the Log Loss cost function. The notebook covers the ML lifecycle using Python, scikit-learn, Pandas, and Matplotlib, from data exploration to model performance analysis. This project showcases a complete end-to-end machine learning workflow, from educational exploration to a powerful, interactive tool for binary classification.*

Click [here](#) to the Live Web App (Hugging Face space).

Click [here](#) to the GitHub Repository (Code & Notebook).

Click [here](#) for a comprehensive deep-dive into the theory and implementation details in my full article titled 'From Probability to Prediction: A Deep Dive into Building and Deploying a Logistic Regression Classifier'.

September 2025 **Multiple Linear Regression: From Theory to Interactive Web App, GitHub & Hugging Face.**

*Skills: Machine Learning, Scikit-Learn, Pandas, NumPy, Matplotlib*

*Description: This project is a comprehensive toolkit for multiple linear regression, delivered through two key components: an educational Jupyter Notebook and a production-ready web application. The web app, deployed on Hugging Face, allows to perform regression analysis on their own CSV data with no coding required. It features a full workflow: data upload, selection of multiple independent variables, a correlation heatmap for feature analysis, model training (with configurable train-test splits), and in-depth evaluation. The evaluation goes beyond standard metrics by including residual analysis to validate model assumptions for a multi-variable context. The Jupyter Notebook offers a guide to the theory and practice behind the app. It explains how multiple linear regression, a supervised learning method, models the relationship between several independent variables ( $x_1, x_2, \dots, x_n$ ) and a continuous target ( $y$ ). The goal is to find the optimal best-fit hyperplane ( $y = \theta_0 + \theta_1 x_1 + \dots + \theta_n x_n$ ) that minimizes the prediction errors (residuals), a process achieved through Ordinary Least Squares (OLS). The notebook covers the ML lifecycle using Python, scikit-learn, Pandas, and Seaborn, including data exploration, model training, and performance analysis with metrics like MAE, MSE, and the  $R^2$  score. This project showcases a complete end-to-end machine learning workflow, from educational exploration to an interactive tool for multivariate analysis.*

Click [here](#) to the Live Web App (Hugging Face space).

Click [here](#) to the GitHub Repository (Code & Notebook).

Click [here](#) for a comprehensive deep-dive into the theory and implementation details in my full article titled 'From Equation to Application: A Deep Dive into Building and Deploying a Multiple Linear Regression Model'.

September 2025 **Simple Linear Regression: From Theory to Interactive Web App, GitHub & Hugging Face.**

*Skills: Machine Learning, Scikit-Learn, Pandas, NumPy, Matplotlib*

*Description: This project is a comprehensive toolkit for simple linear regression, delivered through two key components: an educational Jupyter Notebook and a production-ready web application. The interactive web app, deployed on Hugging Face, allows anyone to perform regression analysis on their own CSV data with no coding required. It features a full workflow: data upload, feature visualization, model training (with configurable train-test splits), and in-depth evaluation. The evaluation goes beyond standard metrics by including residual analysis (Actual vs. Predicted, Residual Plots, and Residual Distribution) to validate model assumptions. The foundational Jupyter Notebook offers a detailed, step-by-step guide to the theory and practice behind the app. It explains how simple linear regression, a supervised learning method, models the relationship between a single independent variable ( $x$ ) and a continuous target ( $y$ ). The goal is to find the optimal best-fit line ( $y = \theta_0 + \theta_1 x$ ) that minimizes the prediction errors (residuals), a process often achieved through Ordinary Least Squares (OLS). The notebook covers the complete ML lifecycle using Python, scikit-learn, and Matplotlib, including data exploration, model training, and performance analysis with metrics like MAE, MSE, RMSE, and  $R^2$  score. This project showcases a complete end-to-end machine learning workflow, from educational exploration to a deployed, interactive tool.*

Click [here](#) to the Live Web App (Hugging Face space).

Click [here](#) to the GitHub Repository (Code & Notebook).

Click [here](#) for a comprehensive deep-dive into the theory and implementation details in my full article titled 'From Equation to Application: A Deep Dive into Building and Deploying a Simple Linear Regression Model'.

## Languages

English **Full professional proficiency.**  
Italian **Native or bilingual proficiency.**

## Computer skills

Application software	<b>MATLAB, Magma</b>	Frameworks	<b>Selenium, Appium, Robot Framework, Katalon, Browser, Playwright</b>
Programming languages	<b>Python, Java, Groovy, C, C++, JavaScript, AutoHotkey</b>	Platforms for API development	<b>Postman</b>
Markup languages	<b>LaTeX, XML, HTML</b>	Version control systems	<b>Git</b>
Integrated Development Environments	<b>PyCharm, IntelliJ Idea, Android Studio, Eclipse, Visual Studio Code, AHK Studio</b>	Software Testing Tools	<b>LoadRunner: VuGen, Controller, Analysis</b>
Operating systems	<b>MacOS, Windows, Linux</b>	Office Suites	<b>Microsoft Office</b>
Behavior-driven developments	<b>Cucumber, Gherkin, User story, Behave</b>	Load Testing Tool	<b>Apache JMeter</b>

Cloud computing service	<b>Microsoft Azure, Google Cloud Platform (GCP)</b>	Issue tracking systems	<b>Jira</b>
Collaborative software	<b>Confluence</b>	Domain-specific languages	<b>SQL</b>
Container orchestration systems	<b>Kubernetes</b>	Document Processing Platform	<b>Hyperscience</b>
Database system	<b>NoSQL</b>	Application Programming Interfaces	<b>REST API</b>
Developer platform	<b>Azure DevOps, GitHub, Hugging Face</b>	CI/CD Execution	<b>GitHub Actions</b>
Test reporting and analytics platforms	<b>Report Portal</b>	Containerization platforms	<b>Docker</b>
Data interchange formats	<b>JSON</b>	Data serialization languages	<b>YAML</b>
Style sheet languages	<b>CSS</b>	Logic programming language	<b>Prolog</b>
Agent-based modeling programming language	<b>NetLogo</b>	Machine Learning libraries	<b>Scikit-learn, Pandas, NumPy, Matplotlib</b>
Automated Planning Algorithms	<b>Graphplan, BlackBox, Fast Forward</b>	Automated Planning language	<b>PDDL</b>
Agent-based modeling IDE	<b>NetLogo</b>		