

PROGRAMMO E INVENTO CON L'IA



*GUIDA ALL'USO DI MACHINE LEARNING FOR KIDS
CON SCRATCH 3.0*

Questa guida è stata realizzata dall'Associazione Dschola per i corsi "Programmo e Invento con l'IA" 2025 con il supporto della Fondazione CRT.

La guida è liberamente scaricabile dal sito dell'Associazione Dschola - <https://www.associazionedschola.it/programmoinvento-ia/le-guide-per-il-coding-a-scuola/>



Quest'opera è distribuita con licenza [Creative Commons CCBY 4.0](https://creativecommons.org/licenses/by/4.0/) Attribuzione Internazionale

Novembre 2024

Saluto del presidente



L'Associazione Dschola, grazie alle competenze dei suoi formatori, a partire dall'anno scolastico 24/25, propone agli studenti del primo biennio delle scuole secondarie di secondo grado, un nuovo percorso formativo ed innovativo, denominato "Intelligenza artificiale con Scratch" che utilizza la piattaforma "Machine Learning for kids" ed il linguaggio di programmazione "Scratch". Si esplorerà la produzione di contenuti multimediali ottenuti attraverso le più note piattaforme per l'intelligenza generativa. Questo corso offrirà agli studenti l'opportunità di avvicinarsi al campo dell'intelligenza artificiale in modo accessibile e divertente, consentendo loro di comprendere i concetti di base del machine learning ed al mondo dei prompt come strumento di affiancamento nello studio, attraverso attività pratiche e progetti stimolanti.

Il Progetto si colloca all'interno della linea didattica-formativa "Programmo e Invento con I.A.", messa in campo dall'Associazione nell'ambito del Progetto Diderot della Fondazione CRT.

Questa guida vuole essere lo strumento necessario ed indispensabile rivolto alle studentesse e agli studenti che potranno allenarsi, con la guida ed il supporto dei formatori, per conseguire le abilità e le competenze necessarie per poter creare progetti e costruire giochi con i modelli di apprendimento automatico che gli stessi hanno già addestrato. La serie di esercitazioni proposte, costituisce un valido pacchetto formativo, in modo tale da privilegiare la dimensione laboratoriale e per consentire agli studenti di esercitarsi nel miglior modo possibile, in sintonia con quanto sostenuto da Diderot con l'espressione: "Giovane, più che istruirti, vorrei esercitarti".

Il Presidente Dschola
Alfonso Lupo

La Fondazione CRT

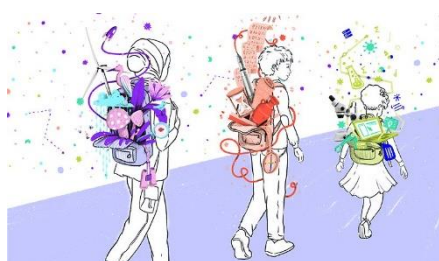


FONDAZIONE CRT

Da oltre 30 anni motore di sviluppo inclusivo e sostenibile

Fondazione CRT, nata nel 1991, è la terza Fondazione di origine bancaria italiana per entità del patrimonio. Ha messo a disposizione del territorio del Nord Ovest più di 2 miliardi di euro, rendendo possibili oltre 42.000 progetti per l'arte e la cultura, la ricerca, la formazione dei giovani, il contrasto alle povertà, il miglioramento della qualità della vita delle persone con disabilità, la tutela dell'ambiente, l'innovazione. Inoltre, ha interamente riqualificato le OGR Torino, trasformandole in un centro internazionale di innovazione tecnologica e culturale.

www.fondazioneCRT.it



IL PROGETTO DIDEROT DELLA FONDAZIONE CRT

Il progetto Diderot di Fondazione CRT propone agli studenti degli Istituti di istruzione primaria e secondaria di I e II grado del Piemonte e della Valle d'Aosta l'opportunità di approfondire le materie tradizionali con metodologie innovative e di avvicinarsi in modo creativo e stimolante a discipline che esulano dallo stretto ambito curriculare, diffondendo i valori fondanti della società civile: dall'arte alla matematica, dall'economia al computing, dagli stili di vita alla prevenzione delle dipendenze, dall'inglese all'arte contemporanea.

Il progetto - che dal 2005 ha formato oltre 1,5 milioni di studenti - è articolato in lezioni, corsi, attività e visite didattiche ed è gratuito per tutte le scuole (escluso il costo di eventuali trasporti).

www.fondazioneCRT.it



@fondazioneCRT



@fondazione-crt



@fondazioneCRT



@FondazioneCRT



@fondazioneCRT

L'Associazione DSchola – Le scuole per le scuole



L'Associazione DSchola. Le scuole per le scuole, si è costituita nel 2004 e da allora è il punto di riferimento per i diversi soggetti operanti in Piemonte interessati all'innovazione nella scuola.

L'Associazione ha operato in questi anni con un modello che ha favorito la condivisione dei saperi e la formazione fra pari: le competenze tecnologiche, le sperimentazioni didattiche e amministrative, sono messe a disposizione di tutte le scuole del territorio attraverso l'organizzazione di seminari e corsi di formazione gratuiti. Per l'assistenza, consulenza e formazione si avvale di tecnici e docenti esperti, in possesso delle necessarie competenze ed appartenenti al mondo della scuola.

L'Associazione ha sperimentato negli anni soluzioni tecnologiche scalabili ed efficaci per la connettività delle classi (modello wireless campus), per la gestione di laboratori informatici e navigazione protetta (Manutenzione zero e Asso DSchola), per l'ospitalità gratuita dei siti delle scuole (Share DSchola), per l'uso di PC in classe (1 computer per ogni studente), per lo sviluppo di comunità di pratiche sull'elearning, sulle mappe mentali e concettuali, sulla media education (DSchola TV). Dal 2012 lavora per la diffusione del coding nella scuola, organizzando iniziative e formazione su Scratch e App Inventor. Dal 2023 esplora e condivide buone pratiche sull'uso dell'Intelligenza artificiale nella didattica.

Per ulteriori informazioni, visitare il sito

www.associazionedschola.it.

Gli autori

Gli autori fanno parte del gruppo di Ricerca e Sviluppo dell'Associazione Dschola

Prof. Luca Pinet, ISILTP di Verrès - Aosta



Docente a tempo indeterminato di Laboratori di scienze e tecnologie informatiche presso l'Istituzione scolastica di istruzione liceale, tecnica e professionale di Verrès, in Valle d'Aosta. Referente, per il territorio della Valle d'Aosta, dell'associazione Dschola. Docente di corsi di formazione per adolescenti per l'introduzione alla programmazione con i linguaggi Scratch e App Inventor, oltre che sul tema della robotica educativa.

Prof. Laura Frasson, ISILTP di Verrès - Aosta



Docente a tempo indeterminato di Informatica presso la scuola superiore ISILTP di Verrès, in Valle d'Aosta. Appassionata di tecnologia, mi piace programmare in vari linguaggi e leggere articoli o guardare video che presentano le nuove applicazioni dell'informatica. Uso quotidianamente gli LLM e i software con AI che sono diventati il mio gioco e aiuto preferito nel lavoro.

Prof. Ing. Claudio Borgogno, ITIS Delpozzo Cuneo

Docente a tempo indeterminato di informatica presso l'ITIS Mario Delpozzo di Cuneo. Appassionato di tecnologia presso il Fablab Cuneo. Ingegnere informatico, progettista di sistemi informativi web ed app mobile. Organizzo corsi sulla intelligenza artificiale e sulle sue applicazioni nel campo didattico. Sono curioso di capire come influirà socialmente questa nuova rivoluzione tecnologica.

Sommario

Saluto del presidente	3
La Fondazione CRT	4
L'Associazione DSchola – Le scuole per le scuole	5
Gli autori.....	6
Questa Guida.....	9
Pillole di Intelligenza artificiale	10
Ambiti applicativi dell'Intelligenza artificiale	10
Machine learning e Deep learning	11
Utilizzare la piattaforma Machine Learning For Kids	12
Creazione di un nuovo progetto	12
Addestramento	13
Impara e testa	15
Programma	16
Scratch: la programmazione per gioco	18
Machine learning con Scratch 3	20
Primo passo: l'addestramento	20
Secondo passo: inserire lo sprite	21
Terzo passo: riconoscimento dei rifiuti	21
Quarto passo: l'indifferenziato.....	22
Quinto passo: la mascotte Kiran	23
Sesto passo: trasformare in quiz	26
Settimo passo: aggiungere i mastelli.....	28
I Chatbot e gli Assistenti intelligenti.	31
Altre applicazioni di Chatbot	32
Gli Assistenti intelligenti in Scratch.....	33
Codificare il progetto senza machine learning.....	33
Addestrare un modello di comandi con il machine learning.	34
Codificare il tuo assistente intelligente con ML	39
Il punteggio di fiducia dell'assistente intelligente.	42
Assistente intelligente con l'immissione vocale.	44
Raccolta dati per migliorare l'assistente intelligente.	45
Glossario dei Blocchi	47

Questa Guida

Questa guida è stata progettata come supporto al corso “Programmo e invento con l’IA” realizzato dall’Associazione DSchola e destinato al biennio delle Scuole secondarie di secondo grado.

La guida fornisce qualche elemento di orientamento al tema dell’intelligenza artificiale e del machine learning e indicazioni passo passo per la realizzazione di semplici applicazioni utilizzando la piattaforma Machinelearningforkids (MLK) e Scratch versione 3.0

Nel capitolo “Pillole di intelligenza Artificiale” sono descritti alcuni termini e concetti utilizzati nei paragrafi seguenti.

Successivamente è descritto come creare un progetto sulla piattaforma MLK e un breve riepilogo del linguaggio Scratch 3.0. Gli studenti potranno cimentarsi con una semplice applicazione basata sul riconoscimento delle immagini utilizzando il machine learning, declinata sul tema della raccolta differenziata e dello sviluppo sostenibile.

La terza parte, dopo una breve introduzione sul tema delle chatbot, introduce l’utilizzo degli assistenti intelligenti nell’ambiente MLK e Scratch 3.0, evidenziando il diverso approccio nella programmazione con e senza il machine learning, guidando gli studenti allo sviluppo di un’applicazione di domotica.

Infine, è incluso un comodo Glossario dei blocchi utilizzabili nella programmazione.

A questa guida è affiancato un’altra guida all’uso dei sistemi di Intelligenza Artificiale Generativa che può essere utilmente integrato nello stesso corso, per lo sviluppo di contenuti digitali. Anche questo secondo volume è liberamente scaricabile dal sito “Programmo e invento con l’IA”.

Pillole di Intelligenza artificiale

In modo semplicistico potremmo definire l'Intelligenza artificiale (abbreviata con IA) come l'abilità di un sistema tecnologico di risolvere problemi o svolgere compiti e attività tipici della mente e delle abilità umane. L'IA è la disciplina che crea macchine, sia hardware che software, in grado di agire autonomamente. In altre parole, l'IA studia come progettare sistemi che possano sembrare intelligenti agli occhi di un osservatore comune.

L'intelligenza artificiale, in base alla sua relazione con l'essere umano, può essere classificata in tre categorie: narrow, general e super.

- **Narrow Artificial Intelligence.** È progettata per eseguire un solo compito specifico in modo molto efficace, spesso superando le capacità umane in quel compito. Tuttavia, se posta in un contesto diverso, questa IA non funziona bene e non può trasferire le sue abilità ad altri campi.
- **General Artificial Intelligence.** Mira a gestire una varietà di compiti intellettuali imitando il pensiero umano. L'obiettivo è creare un sistema che possa pensare e imparare come una persona.
- **Super Artificial Intelligence.** È ipotetica e rappresenta un'intelligenza che supera di gran lunga le capacità umane in tutti i settori, dalla creatività alla risoluzione dei problemi e alla saggezza generale.

L'intelligenza artificiale, in base alla sua modalità di ragionamento, si divide in due tipi: debole (weak) e forte (strong).

- **Weak Artificial Intelligence.** Simula alcune funzioni cognitive umane senza avere una vera intelligenza. Si basa sul "come se", cioè agisce come se fosse intelligente, ma non lo è realmente. Un esempio è un programma di scacchi: non ha coscienza di sé, ma gioca bene a scacchi.
- **Strong Artificial Intelligence.** Mira a sviluppare una propria intelligenza, non solo a imitare quella umana. Se programmata correttamente, può diventare una mente con capacità cognitive indistinguibili da quelle umane.

Ambiti applicativi dell'Intelligenza artificiale

1. **eHealth:** Diagnosi mediche, analisi di immagini, pianificazione dei trattamenti e monitoraggio dei pazienti. Trattamenti su misura basati sul profilo genetico e sulla storia clinica del paziente.
2. **ChatBot:** Assistenti virtuali che dialogano attraverso voce o testo con gli esseri umani.
3. **eCommerce:** Raccomandazioni di prodotti, ottimizzazione delle catene di fornitura, assistenti virtuali per clienti.
4. **Automobili Autonome:** Guida autonoma, sistemi di assistenza alla guida, gestione del traffico.
5. **Agricoltura:** Monitoraggio delle colture, previsione dei rendimenti, gestione delle risorse.
6. **Industria 5.0:** Ottimizzazione dei processi produttivi, manutenzione predittiva, robotica industriale.

7. **eLearning**: Personalizzazione dell'apprendimento, tutoraggio automatizzato, analisi delle prestazioni degli studenti.
8. **Smart Home**: Automazione domestica, assistenti vocali, gestione energetica. Collegamento di dispositivi intelligenti IoT per creare ambienti automatizzati.
9. **Riconoscimento biometrico**: Sorveglianza, diagnostica medica, analisi di sicurezza. Riconoscimento vocale, riconoscimento facciale, intelligenza sensoriale.
10. **IA Generativa**: Creazione automatica di contenuti complessi, come testi, immagini, video e musica. Traduzione automatica, sottotitolazione, assistenza linguistica.
11. **Servizi di Reclutamento**: Screening dei candidati, analisi delle competenze, pianificazione delle carriere.
12. **Pubblica sicurezza**: Rilevamento di intrusioni, protezione dei dati, analisi delle vulnerabilità. Utilizzo di IA per prevedere e gestire catastrofi naturali e crisi umanitarie. Sistemi avanzati di sorveglianza e analisi dei comportamenti.

Machine learning e Deep learning

Il **machine learning** è un ramo dell'intelligenza artificiale in cui i computer, attraverso un apprendimento automatico, imparano attraverso l'esperienza a fare cose imitando il modo in cui impariamo noi umani.

Il computer impara associando input (**dati in ingresso**) a output (**risultati desiderati**). Si forniscono al computer molti esempi di input e output, quindi esso analizza questi dati per trovare una regola che colleghi gli input agli output: fase di **addestramento**. Durante il training, il computer sviluppa una funzione matematica (**funzione obiettivo**) che può adattarsi a nuovi dati simili a quelli su cui è stato addestrato. Man mano che il computer riceve più dati diventa più bravo a fare le sue previsioni o a compiere le sue azioni.

Questo processo di apprendimento avviene in diversi modi.

- **Supervised Learning** (Apprendimento Supervisionato). Il computer impara da esempi già risolti. In pratica, al computer vengono forniti i dati di ingresso e i risultati attesi, e lui deve trovare la funzione obiettivo.
- **Unsupervised Learning** (Apprendimento Non Supervisionato). Il computer cerca di trovare schemi nascosti nei dati senza sapere in anticipo cosa cercare. In pratica, al computer vengono forniti solo i dati di ingresso, e lui deve trovare la funzione obiettivo e i risultati attesi.
- **Semi-Supervised Learning** (Apprendimento Semi-Supervisionato). Combina gli approcci supervisione e non supervisione, utilizzando alcuni dati etichettati e altri no.c
- **Reinforcement Learning** (Apprendimento per Rinforzo): Il computer impara interagendo con un ambiente e ricevendo premi o punizioni in base alle sue azioni.

Il **deep learning** è una tipologia avanzata di machine learning che permette ai computer di imparare e prendere decisioni imitando il funzionamento del cervello umano. Funziona utilizzando reti neurali artificiali, che sono composte da molti strati di nodi che elaborano le informazioni in modo simile ai neuroni del cervello.

Utilizzare la piattaforma Machine Learning For Kids

Per la realizzazione di software che sfrutta le tecniche del machine learning, e quindi dell'intelligenza artificiale, in questo manuale utilizzeremo la piattaforma **Machine learning for kids**. *Machine learning for kids* è una piattaforma dedicata all'apprendimento del machine learning.

Creazione di un nuovo progetto

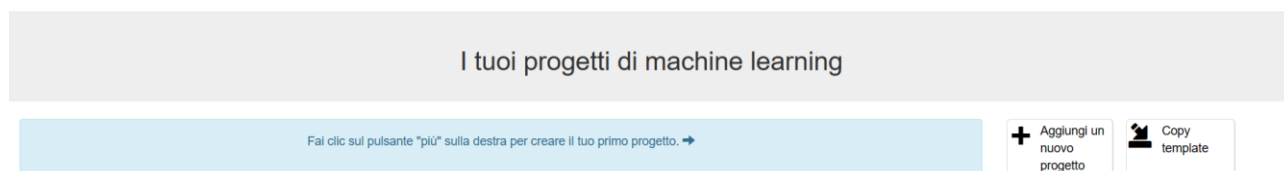
Qui avrete l'opportunità di esplorare in modo approfondito un piccolo ramo dell'intelligenza artificiale, dove le macchine possono apprendere e prendere decisioni intelligenti. La piattaforma dà la possibilità di allenarla nella catalogazione delle immagini e del testo.

Per iniziare, visitate il sito <https://machinelearningforkids.co.uk> e cliccate sul pulsante blu "Iniziamo".

Sarà necessario effettuare l'accesso, quindi avrete le seguenti opzioni:

- **Iscriviti:** Selezionate questa opzione al vostro primo accesso per registrare un account e generare una password. Inserite i vostri dati personali, come nome utente ed email. Sugeriamo di registrarsi come "Insegnante" con "Account libero" per massimizzare le opportunità di sviluppo. La medesima registrazione è da far effettuare agli studenti.
- **Accedi:** Utilizzate questa opzione per accedere al vostro account, previa registrazione e possesso di username e password.
- **Prova subito:** Se volete semplicemente esplorare la piattaforma e testarne le funzionalità senza creare un account, potete selezionare questa opzione. Tuttavia, tenete presente che senza un account, i vostri progressi verranno persi una volta che uscite dal sito.

Una volta effettuato l'accesso, vi troverete nella schermata "I tuoi progetti di machine learning". Per creare un nuovo progetto, cliccate su "Aggiungi un nuovo progetto".



Successivamente, inserite i seguenti dati:

- **Nome del progetto:** Scegliete un nome significativo che rappresenti il vostro progetto.
- **Tipo di progetto:** Selezionate il tipo di dato che desiderate catalogare. Ad esempio, se volete distinguere tra vari tempi verbali, selezionate "Riconoscimento testo". Se invece volete classificare diverse immagini, scegliete "Riconoscimento immagini".

Una volta inseriti i dati del progetto, cliccate su "Crea".

Successivamente, selezionate il progetto desiderato, che vi darà accesso a tre differenti opzioni. Iniziate con **"Addestramento"** per proseguire con la vostra avventura nell'apprendimento del machine learning.



Addestramento

La *catalogazione* nel machine learning è un processo fondamentale che consiste nel classificare e organizzare dati in categorie o classi specifiche, al fine di rendere più efficiente l'analisi e l'elaborazione delle informazioni. Questo processo è cruciale in molte applicazioni del machine learning, come il riconoscimento di immagini, la classificazione di testi, la raccomandazione di prodotti e molto altro.

Per comprendere meglio la catalogazione nel contesto del machine learning, possiamo esaminare un esempio pratico come il riconoscimento facciale. Immaginiamo di avere un vasto insieme di immagini contenenti volti umani e il nostro obiettivo è quello di identificare automaticamente le persone presenti in queste immagini. La prima fase di questo processo è la catalogazione, che coinvolge la suddivisione delle immagini in categorie o classi, come ad esempio "persone" e "non persone".

Per eseguire la catalogazione, possiamo utilizzare algoritmi di machine learning supervisionati o non supervisionati. Nei *sistemi supervisionati*, addestreremo un modello utilizzando un insieme di dati di addestramento etichettati, ovvero dati in cui è già nota l'appartenenza a una determinata classe. Il modello imparerà quindi a riconoscere i pattern nei dati e a associarli alle categorie corrette. Nei *sistemi non supervisionati*, invece, il modello cercherà di identificare pattern o strutture nei dati senza l'ausilio di etichette predefinite.

Nel caso della piattaforma *Machine learning for kids* l'addestramento dell'intelligenza artificiale avviene attraverso l'utilizzo di un sistema supervisionato. Sarete quindi voi sia a stabilire i nomi delle etichette, sia ad inserire il testo e le immagini associati ad ogni etichetta.

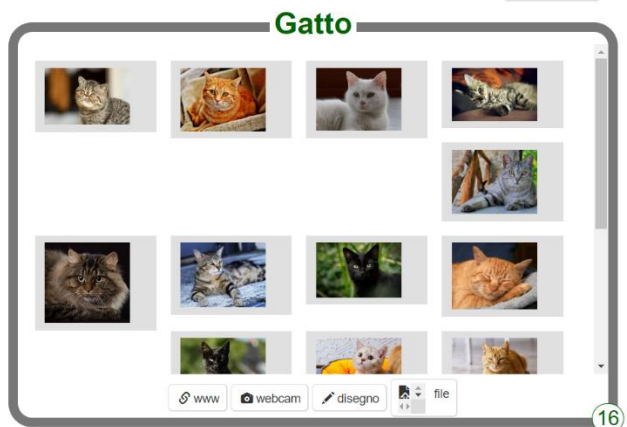
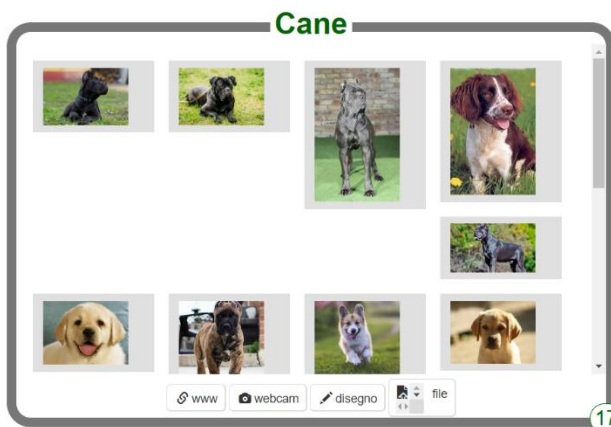
Per iniziare la fase di addestramento è necessario stabilire delle etichette con cui catalogare gli elementi. Per inserire una nuova etichetta cliccare su **"Aggiungi nuova etichetta"**.

Successivamente, è necessario assegnare all'etichetta un nome. Dopo aver creato l'etichetta, è possibile inserire i dati con diverse modalità di input. Se la catalogazione è fatta tra immagini:

1. Inserimento di un'immagine usando il suo collegamento ipertestuale (**www**);
2. Inserimento di un'immagine catturata usando la **webcam**;
3. Creazione di un'immagine tramite uno strumento di **disegno**;
4. Inserimento di un'immagine locale dal proprio computer (**file**).
5. Drag and drop di un'immagine da una pagina del browser.

[< Ritorna ai progetti](#)

+ Aggiungi una nuova etichetta



Al contrario, se la catalogazione è fatta tra **testi**, sarà possibile cliccare sulla voce “aggiungi un esempio” per inserire parola dopo parola, o frase dopo frase.

[< Ritorna ai progetti](#)

+ Aggiungi una nuova etichetta



Alcuni esempi di esercizi che possono essere proposti sulla catalogazione delle immagini sono:

- Classificare immagini che distinguono un cane da un gatto;
- Classificare immagini che distinguono un essere vivente in base al suo regno;
- Classificare immagini che distinguono un testo da corsivo a stampatello;
- Classificazione di immagini di un materiale in stato solido, liquido e gassoso;
- Classificazione delle immagini sulle varie tipologie di stelle.

Al contrario, alcuni esempi di esercizi che possono essere proposti sulla catalogazione del testo sono:

- Classificare verbi in indicativo, congiuntivo e condizionale
- Classificare le parole in base alla lingua utilizzata (multiculturalità)
- Riconoscere le parole d’odio all’interno di un articolo di giornale o un testo
- Riconoscere le mail di spam all’interno di una casella di posta elettronica
- Classificazione degli esseri viventi nei cinque regni

Impara e testa

Una volta che il modello è stato addestrato, possiamo utilizzarlo per catalogare nuovi dati, ossia per assegnare loro una classe o categoria appropriata. Ad esempio, se vogliamo identificare le persone nelle nuove immagini, utilizzeremo il modello addestrato per predire se i volti presenti appartengono o meno alla classe "persone".

La catalogazione nel machine learning è un processo iterativo che richiede un'attenta progettazione, raccolta dei dati, preparazione dei dati, selezione degli algoritmi appropriati e valutazione dei modelli. È importante notare che la qualità della catalogazione dipende in gran parte dalla qualità dei dati utilizzati durante il processo di addestramento del modello. Pertanto, è fondamentale garantire la disponibilità di dati accurati e rappresentativi per ottenere risultati affidabili e precisi.

Per poter procedere nella fase di **"Impara & testa"**, disponibile dal bottone **"Ritorna ai progetti"**, è necessario che in fase di **"Addestramento"** siano stati aggiunti almeno sette esempi per ciascuna etichetta. Tuttavia, più esempi fornirai ad ogni etichetta, più il riconoscimento di nuove immagini appartenenti a quell'etichetta sarà accurato.

Cos'hai fatto?

Hai raccolto esempi di immagini che un computer può utilizzare per riconoscere quando le immagini sono Cane or Gatto.

Hai raccolto:

- 17 examples of Cane,
- 16 examples of Gatto

E adesso che succede?

Pronto per avviare l'addestramento?

Fai clic sul pulsante in basso per iniziare ad addestrare un modello di apprendimento automatico usando gli esempi che hai raccolto finora

(Oppure torna alla pagina [Addestramento](#) se vuoi prima raccogliere altri esempi.)

Informazioni dal computer di addestramento:

Addestra nuovi modelli di machine learning

Arrivati a questo punto, per dare in pasto tutti gli esempi caricati all'algoritmo di machine learning, è necessario premere il pulsante **"Addestra nuovi modelli di machine learning"**.

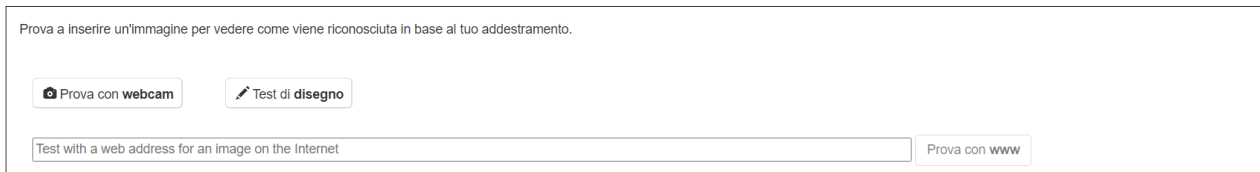
Riassumendo, quando un modello di machine learning viene addestrato per stabilire un algoritmo di classificazione per nuovi elementi, il processo inizia con il caricamento dei dati inseriti durante la fase di addestramento. Questi dati sono fondamentali poiché il modello impara dai pattern presenti in essi per poter fare previsioni accurate sui nuovi dati.

Una volta caricati i dati, il modello di machine learning passa alla fase di addestramento effettivo, durante la quale cerca di individuare i pattern e le relazioni tra i dati. Questo avviene attraverso l'applicazione di un algoritmo di apprendimento specifico, che può variare in base al tipo di problema di classificazione e ai dati disponibili.

Durante il processo di addestramento, il modello regola i suoi parametri interni in modo da minimizzare l'errore tra le previsioni effettuate e le etichette corrette associate ai dati di addestramento. Questo processo di ottimizzazione può richiedere molte iterazioni.

Una volta completata la fase di addestramento, il modello è pronto per fare previsioni sui nuovi dati. Quando viene presentato un nuovo elemento da classificare, il modello utilizza gli algoritmi e i parametri appresi durante l'addestramento per determinare la classe o la categoria appropriata per quel dato.

In breve, il modello di machine learning carica i dati inseriti durante la fase di addestramento per stabilire un algoritmo di classificazione per nuovi elementi attraverso un processo di apprendimento dei pattern nei dati e di ottimizzazione dei parametri interni del modello. Una volta addestrato, il modello è in grado di fare previsioni accurate sui nuovi dati utilizzando gli algoritmi e i parametri appresi durante l'addestramento.

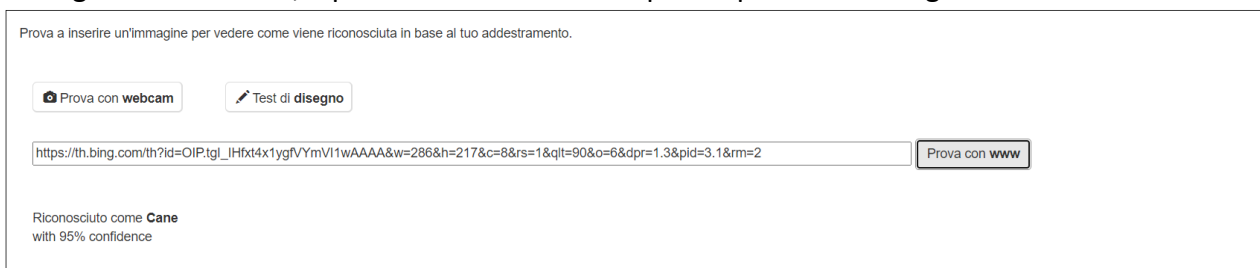


Prova a inserire un'immagine per vedere come viene riconosciuta in base al tuo addestramento.

Prova con webcam Test di disegno

Test with a web address for an image on the Internet Prova con www

Per poter testare il modello è possibile, nel caso della catalogazione delle immagini, usare tre modalità differenti: prova con webcam, test di disegno e prova con www. Nel caso, invece, della catalogazione del testo, è possibile inserire nuove parole per testare l'algoritmo.



Prova a inserire un'immagine per vedere come viene riconosciuta in base al tuo addestramento.

Prova con webcam Test di disegno

https://th.bing.com/th?id=OIP.tgl_IHfxt4x1ygfVYmV1wAAAA&w=286&h=217&c=8&rs=1&qit=90&o=6&dpr=1.3&pid=3.1&rm=2 Prova con www

Riconosciuto come **Cane**
with 95% confidence

Inserito un nuovo esempio, la piattaforma restituirà due dati molto importanti:

La categoria di appartenenza dell'esempio fornito secondo l'algoritmo.

La percentuale di **confidence**, ovvero la percentuale di affidabilità del risultato fornito.

Nel caso in cui, apportata una serie di esempi, essi vengono riconosciuti correttamente e con una percentuale di confidence sufficientemente alta, allora l'addestramento è avvenuto con successo. Altrimenti, sarà necessario caricare nuovi esempi di addestramento.

Programma

Una volta terminate le fasi di “**Addestramento**” e di “**Impara & testa**”, è possibile accedere alla voce “**Programma**”.

Scratch 3

Usa la nuova versione di Scratch



Scratch 3



Python

Scrivi il codice Python per utilizzare il tuo modello di apprendimento automatico



Python



App Inventor

Crea un'applicazione mobile per il tuo telefono o tablet



App Inventor



Arrivati a questa fase, è possibile scegliere il linguaggio di programmazione da utilizzare, con integrate le istruzioni del modello addestrato. Tra i linguaggi proposti ci sono:

- **Scratch.** Linguaggio a blocchi adatto ai non esperti nella programmazione.
- **App Inventor.** Linguaggio a blocchi per creare app nel sistema operativo Android.
- **Python.** Linguaggio di programmazione ad alto livello.

In questo manuale verrà utilizzato, per la realizzazione di software che sfruttano il machine learning, il linguaggio a blocchi Scratch.

Scratch: la programmazione per gioco

Scratch è un ambiente grafico di lavoro per imparare lo sviluppo di programmi in modo semplice e creare storie interattive, giochi, animazioni grafiche, simulazioni e altro ancora, per poi condividere i programmi realizzati attraverso il Web.

La codifica dei programmi in Scratch consiste nell'impilare blocchi, che presentano forma e colore dipendenti dall'istruzione che si vuole utilizzare, come si fa con i mattoncini delle costruzioni, pezzo dopo pezzo. Così facendo, è possibile avvicinarsi alla programmazione e capirne la logica alla base in modo divertente e creativo.

I programmi di Scratch agiscono su oggetti grafici, disegni, immagini chiamati sprite, come la figurina del gatto che rappresenta l'icona di Scratch. E' possibile disegnare gli sprite a piacere attraverso un semplice programma di disegno, così come è possibile importare un'immagine o una foto scattata con una macchina fotografica digitale o con la webcam. Gli sprite, inoltre, possono essere personalizzati associando costumi diversi, in modo da animarli dando loro la forma che più interessa, e suoni diversi. A ogni sprite sono associati uno o più script, insieme di blocchi-istruzioni che indicano allo sprite che cosa deve fare: parlare, muoversi, suonare, nascondersi, mostrarsi, eseguire calcoli e tanto altro ancora.

L'ambiente è suddiviso fondamentalmente in tre aree diverse:

- a destra della videata troviamo lo **stage**, il palcoscenico, l'area in cui gli sprite interagiscono durante l'esecuzione del programma, che parte di solito quando si fa clic sul pulsante con la bandierina verde, posto sopra lo stage; gli sprite sono dei disegni o delle immagini, presi tra quelli messi a disposizione da Scratch o creati da noi, con cui si interagisce quando è lanciato il programma e che eseguono i blocchi (le istruzioni) che costituiscono lo script (il programma);
- alla sinistra troviamo l'elenco dei **blocchi** che servono a codificare il programma, suddivisi in 8 categorie caratterizzate ciascuna da un diverso colore; sono in pratica i mattoncini che si utilizzano per costruire il programma impilandoli nella sequenza desiderata;
- al centro, infine, vi è l'area per programmare, detta **area script**, in cui trasciniamo e impiliamo con la tecnica del drag and drop i blocchi che servono alla codifica del programma.

Lo **sprite** è l'oggetto posto nello stage a cui vengono associati uno o più script per farlo muovere, interagire, calcolare, giocare, e altro ancora. Oltre allo sprite raffigurante il gatto, esistono molti altri sprite già disponibili nel sito, oppure è possibile crearne dei nuovi attraverso un apposito **editor**.

Agendo su questi tre pulsanti, posti al di sotto dello stage, è possibile:

- scegliere uno sprite dalla libreria;
- disegnare un nuovo sprite;
- caricare uno sprite da file;
- creare un nuovo sprite dalla webcam.


I **blocchi** sono suddivisi in 10 categorie di colore diverso in base al tipo di azione che effettuano. Nelle categorie **Movimento**, **Aspetto**, **Suono** si trovano i blocchi che permettono di applicare allo sprite le animazioni e i suoni desiderati; nella categoria **Penna** si trovano i blocchi per sviluppare programmi che permettono di scrivere e disegnare; i blocchi delle categorie **Controllo**, **Operatori**, **Variabili e Liste** permettono di codificare le principali strutture algoritmiche (se... allora... altrimenti, ripeti finì a quando..., ripeti... volte, ecc.) che agiscono sulle variabili; la categoria **Situazioni** contiene i blocchi per la gestione degli eventi che lanciano l'esecuzione degli script; nella categoria **Sensori**

troviamo i blocchi per testare il verificarsi di certe situazioni; infine, nella categoria **Altri Blocchi** troviamo raccolti i nuovi blocchi che possiamo creare associando loro uno script in modo da gestire i sottoprogrammi.

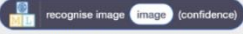
Nel caso specifico del linguaggio Scratch legato alla piattaforma *Machine Learning for kids*, vengono riportati di seguito i blocchi di istruzione utilizzabili nella realizzazione del codice.

[Apri in Scratch 3](#)


Your project will add these blocks to Scratch.



Inserire images nell'input e restituirà l'etichetta con cui il modello di apprendimento automatico lo riconosce.

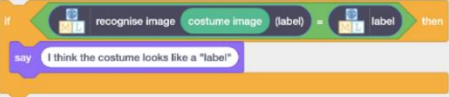


Ciò restituirà la sicurezza del modello di apprendimento automatico che riconosce il tipo di images. (Come un numero compreso tra 0 e 100).

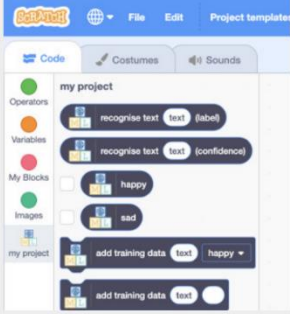


Questi blocchi rappresentano le etichette che hai creato nel tuo progetto, quindi puoi usare i loro nomi negli script.

Ciò significa che puoi fare qualcosa del genere:



Assomiglierà a qualcosa del genere - tranne per il nome del tuo progetto.



Machine learning con Scratch 3

In questo capitolo vedremo come realizzare un'applicazione che utilizza il machine learning per un progetto di riconoscimento automatico delle immagini, declinato sul tema della raccolta differenziata dei rifiuti.

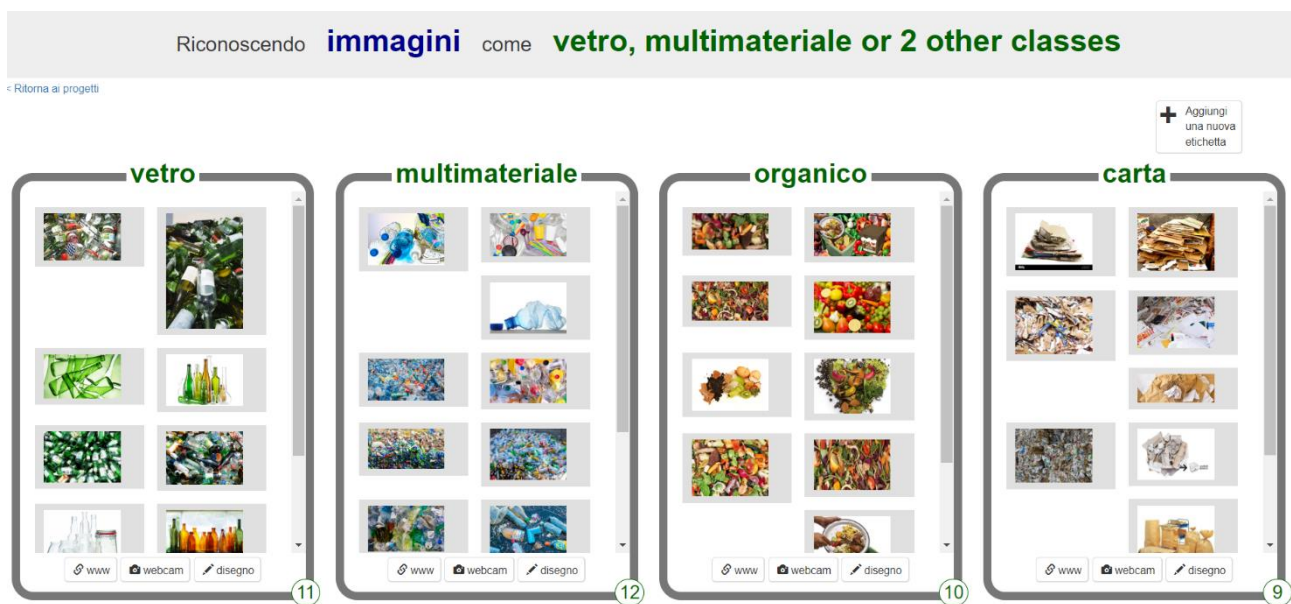
Primo passo: l'addestramento

Partendo dalla schermata "I tuoi progetti di machine learning". Per creare un nuovo progetto, cliccate su "Aggiungi un nuovo progetto".

Nella voce "Nome del progetto" digitare il titolo "Raccolta differenziata". Nella voce "Tipo di progetto" selezionare "Riconoscimento immagini". Una volta inseriti i dati del progetto, cliccate su "Crea".

Successivamente, selezionate il progetto "Raccolta differenziata", che vi darà accesso a tre differenti opzioni. Iniziate con "Addestramento".

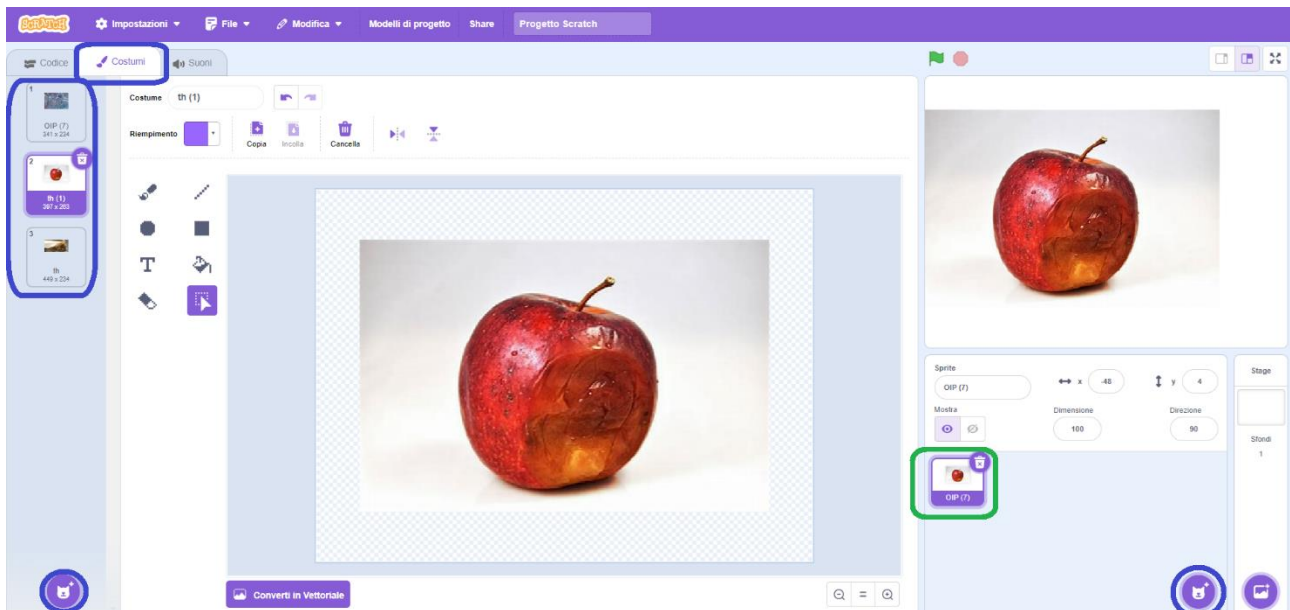
Per iniziare la fase di addestramento è necessario stabilire delle etichette con cui catalogare gli elementi. Per inserire una nuova etichetta cliccare su "Aggiungi nuova etichetta". Le etichette che è necessario creare sono quattro: "Multimateriale", "Organico", "Vetro" e "Carta". Per ogni etichetta è necessario inserire il maggior numero possibile di immagini (il numero minimo è di sette immagini per ogni etichetta).



A questo punto è possibile procedere con la fase di "Impara & testa", disponibile dal bottone "Ritorna ai progetti". Per dare in pasto tutti gli esempi caricati all'algoritmo di machine learning, è necessario premere il pulsante "Addestra nuovi modelli di machine learning".

Una volta che, date nuove immagini alla piattaforma, questa riuscirà a riconoscerle con un discreto livello di confidenza, allora sarà possibile accedere alla voce "Programma" e selezionare tra i linguaggi di programmazione proposti "Scratch" e poi cliccare "Apri in Scratch 3".

Secondo passo: inserire lo sprite



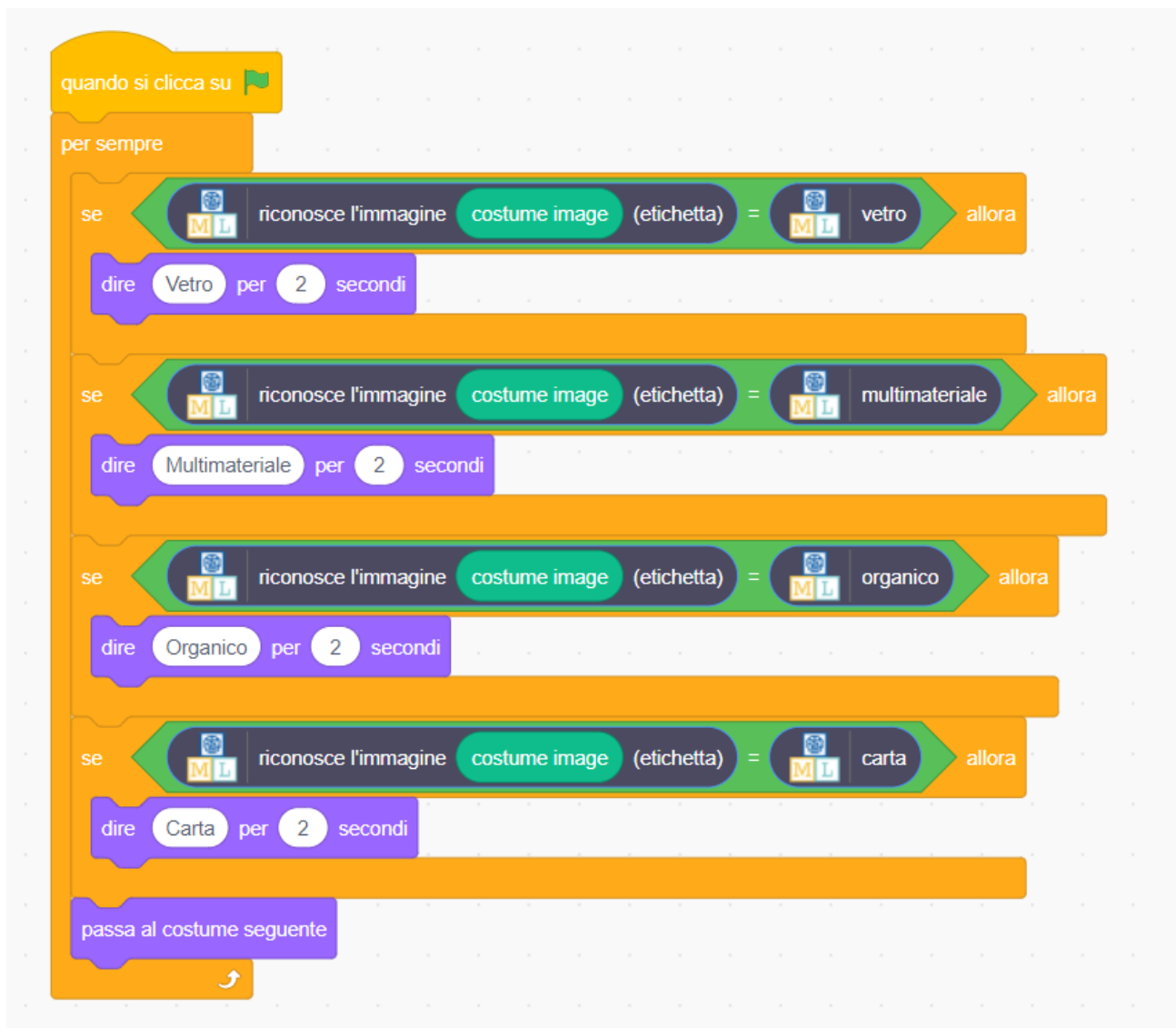
Per prima cosa è necessario inserire un nuovo Sprite (in basso a destra) e per questo nuovo Sprite inserire una serie di Costumi (in basso a sinistra). I costumi sono una serie di rifiuti da gettare, e devono (anche se non obbligatoriamente) essere delle immagini differenti rispetto a quelle utilizzate per addestrare il modello.

Importante: Spesso si fa l'errore di inserire ogni singolo rifiuto come sprite. Lo Sprite rappresenta complessivamente tutti i rifiuti che vanno gettati nella pattumiera. I singoli Costumi sono i singoli rifiuti che vanno gettati. Quindi tutti i rifiuti vanno inseriti in un unico sprite.

Terzo passo: riconoscimento dei rifiuti

Come passo successivo è necessario associare allo Sprite i rifiuti da riconoscere.

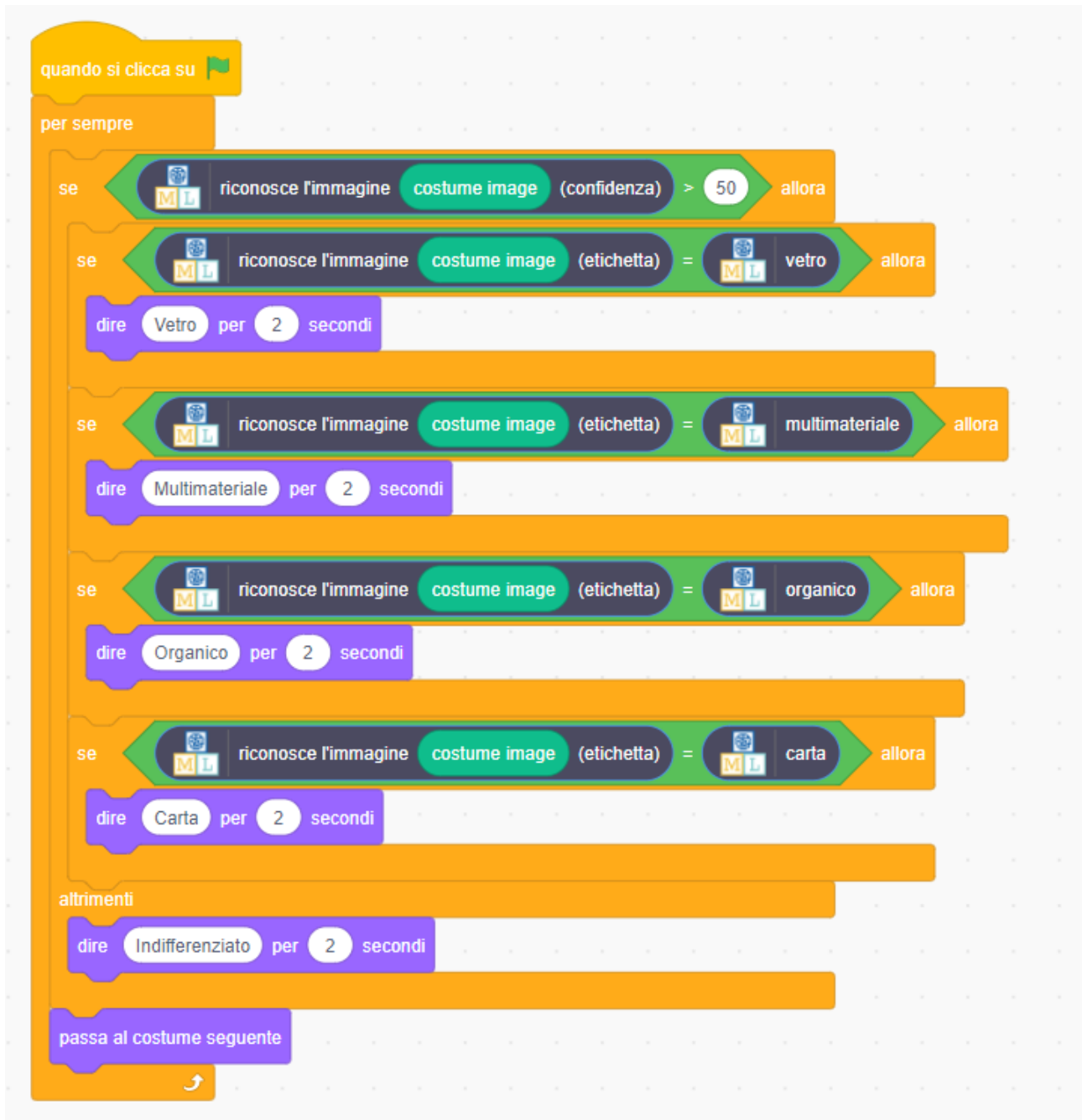
Di seguito il codice perché riconosca il tipo di rifiuto (costume) e dica in quale bidone deve essere gettato.



In questo primo esempio la classificazione viene fatta solo tra vetro, multimateriale, organico e carta, ovvero le quattro etichette che erano state inserite in fase di Addestramento.

Quarto passo: l'indifferenziato

La logica che si utilizza per l'aggiunta del bidone dell'indifferenziato è che tutto ciò che non va gettato negli altri bidoni è da considerare indifferenziato. Ovvero, se un rifiuto non rientra né nella categoria del vetro, né in quella del multimateriale, né in quella dell'organico, né in quella della carta, allora è da gettare nell'indifferenziato.



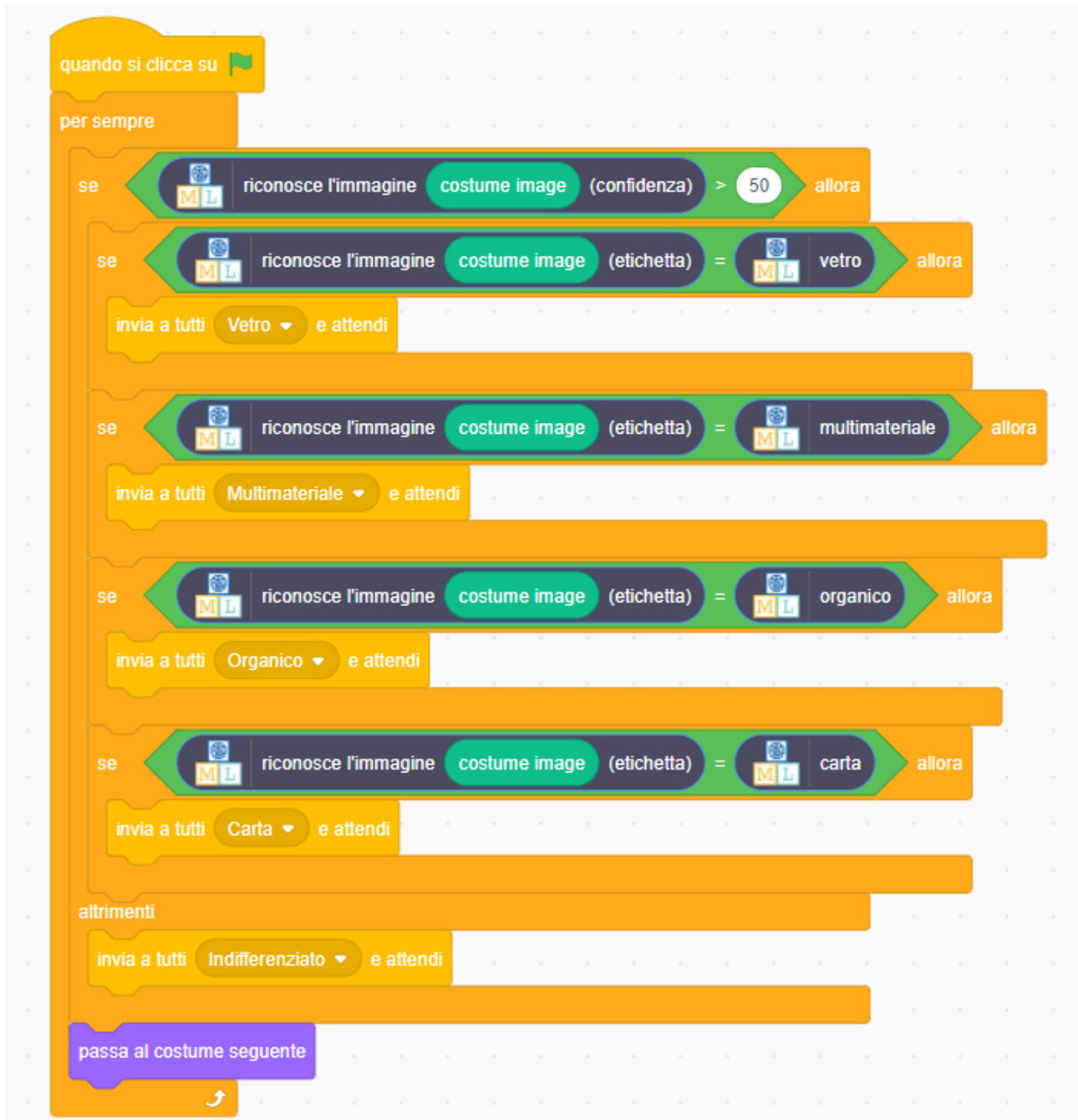
Per fare questo, come indicato nel codice, si deve utilizzare la confidenza. Ovvero, se la confidenza di una certa associazione tra il costume e l'etichetta risulta troppo debole, allora si considera il rifiuto come non catalogato e lo si getta nell'indifferenziato.

Quinto passo: la mascotte Kiran

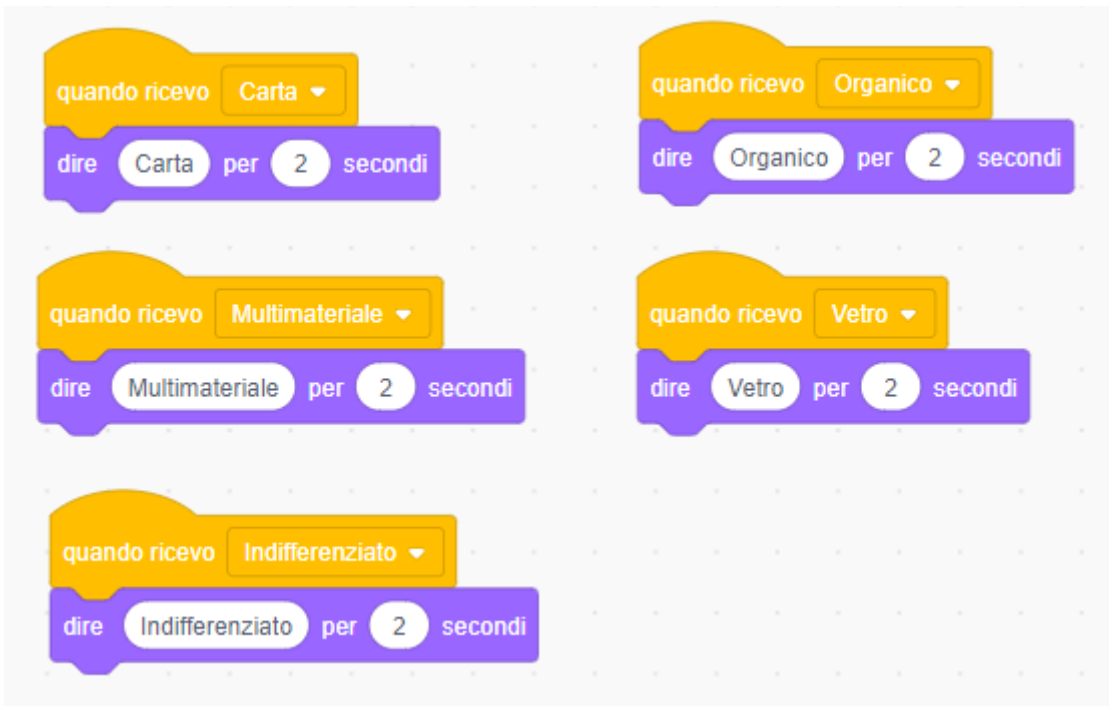
Per rendere il gioco più divertente, possiamo aggiungere la mascotte Kiran, che ci dirà in quale bidone va gettato il rifiuto. In pratica, invece di chiedere al rifiuto di auto-catalogarsi, lo facciamo catalogare ad un nuovo Sprite che si occuperà di dialogare con l'utente.



Per fare questo, è necessario modificare il codice dello Sprite Rifiuto facendo sì che non dica più quale tipo di rifiuto è, ma che scambi un messaggio con Kiran.



Allo stesso tempo, lo Sprite Kiran dovrà associare ad ogni messaggio ricevuto alcune istruzioni da eseguire, ovvero la catalogazione del rifiuto. In pratica verranno individuati cinque messaggi: carta, organico, multimateriale, vetro, indifferenziato. In base al messaggio che lo sprite Rifiuto invia, e che di conseguenza lo sprite Kiran riceve, verrà stampato sullo schermo la tipologia di rifiuto in questione.

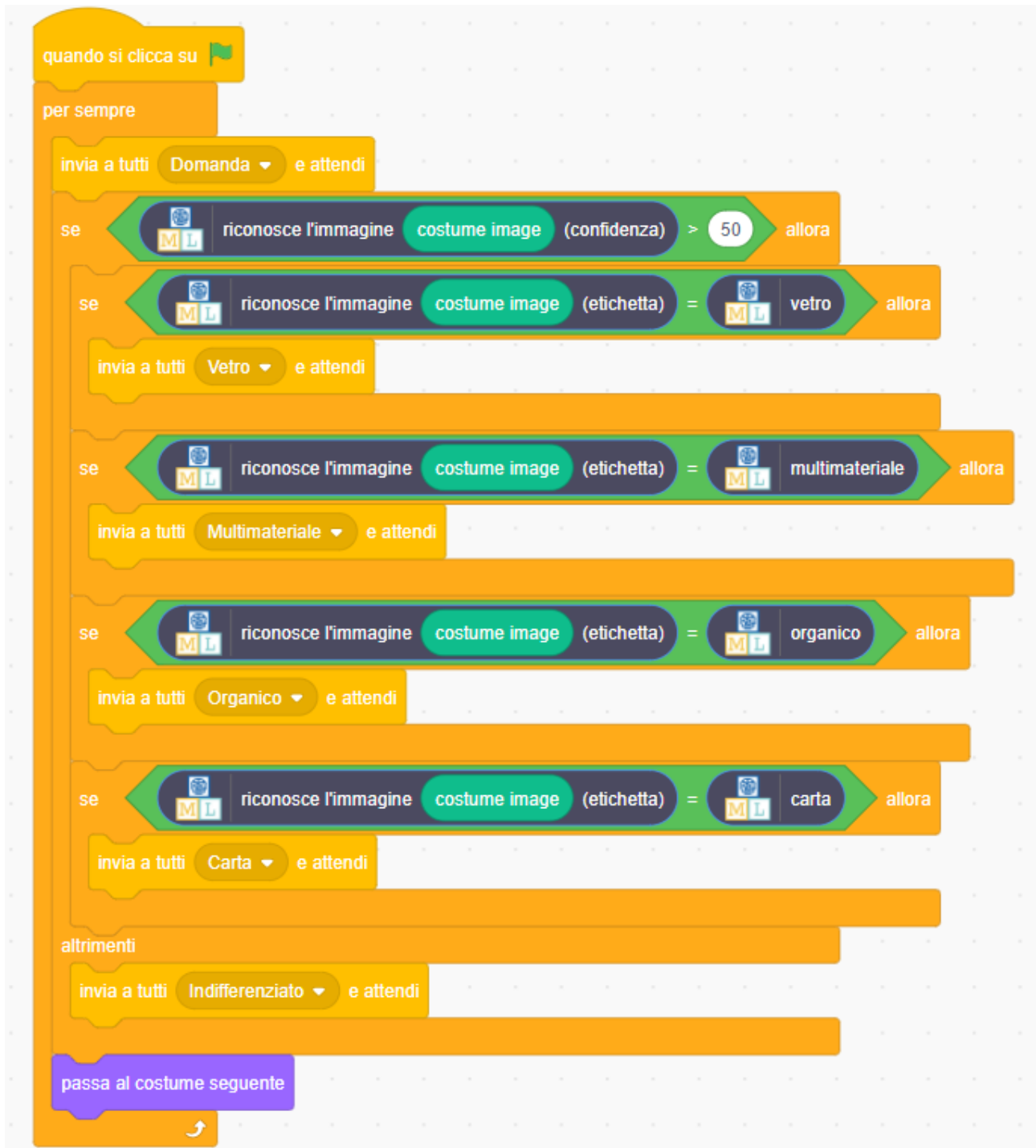


Sesto passo: trasformare in quiz

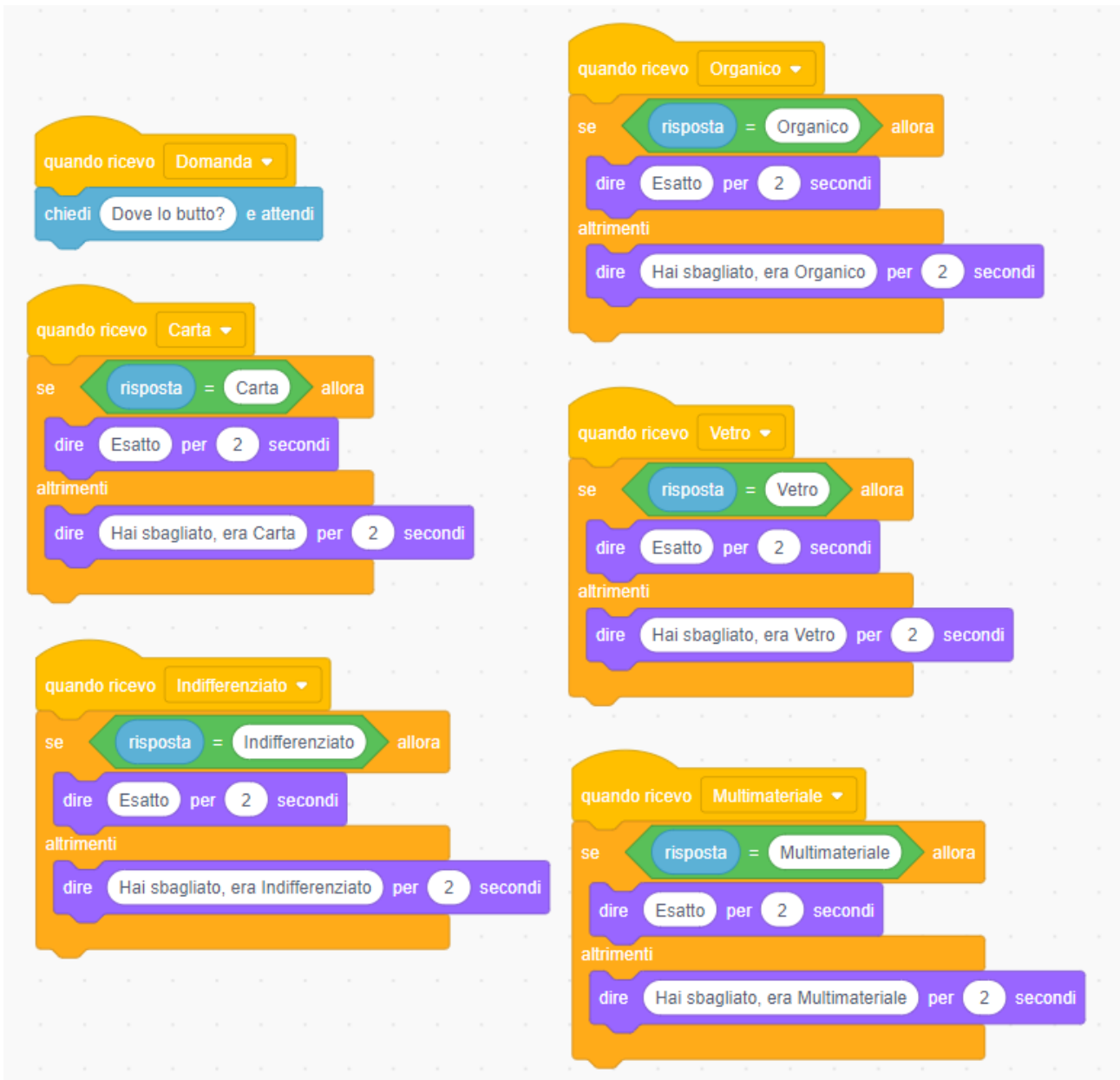
Per aumentare sempre di più il livello del gioco e per renderlo più interattivo coinvolgendo in maniera diretta il giocatore, è possibile far sì che Kiran non dica più egli stesso immediatamente all'utente dove va gettato il rifiuto, ma che al contrario faccia dire al giocatore dove va buttato, sotto forma di quiz.

In pratica, ad ogni rifiuto, Kiran dovrà chiedere al giocatore "Dove lo butto?" e il giocatore dovrà inserire tramite tastiera, all'interno della casella di testo che gli si aprirà, la risposta. Successivamente, Kiran dovrà dire al giocatore se la risposta che ha inserito era corretta con "Esatto!", oppure se non lo era con "Hai sbagliato" e in tal caso aiutarlo a gettare il rifiuto nel contenitore corretto.

Per fare questo, è necessario che all'inizio dello sprite Rifiuto si aggiunga l'invio del messaggio Domanda a Kiran, così che Kiran capisca di dover effettuare la domanda.



Di conseguenza, il codice dello sprite Kiran andrà completamente trasformato, per far sì che effettui la domanda e ne verifichi la risposta.



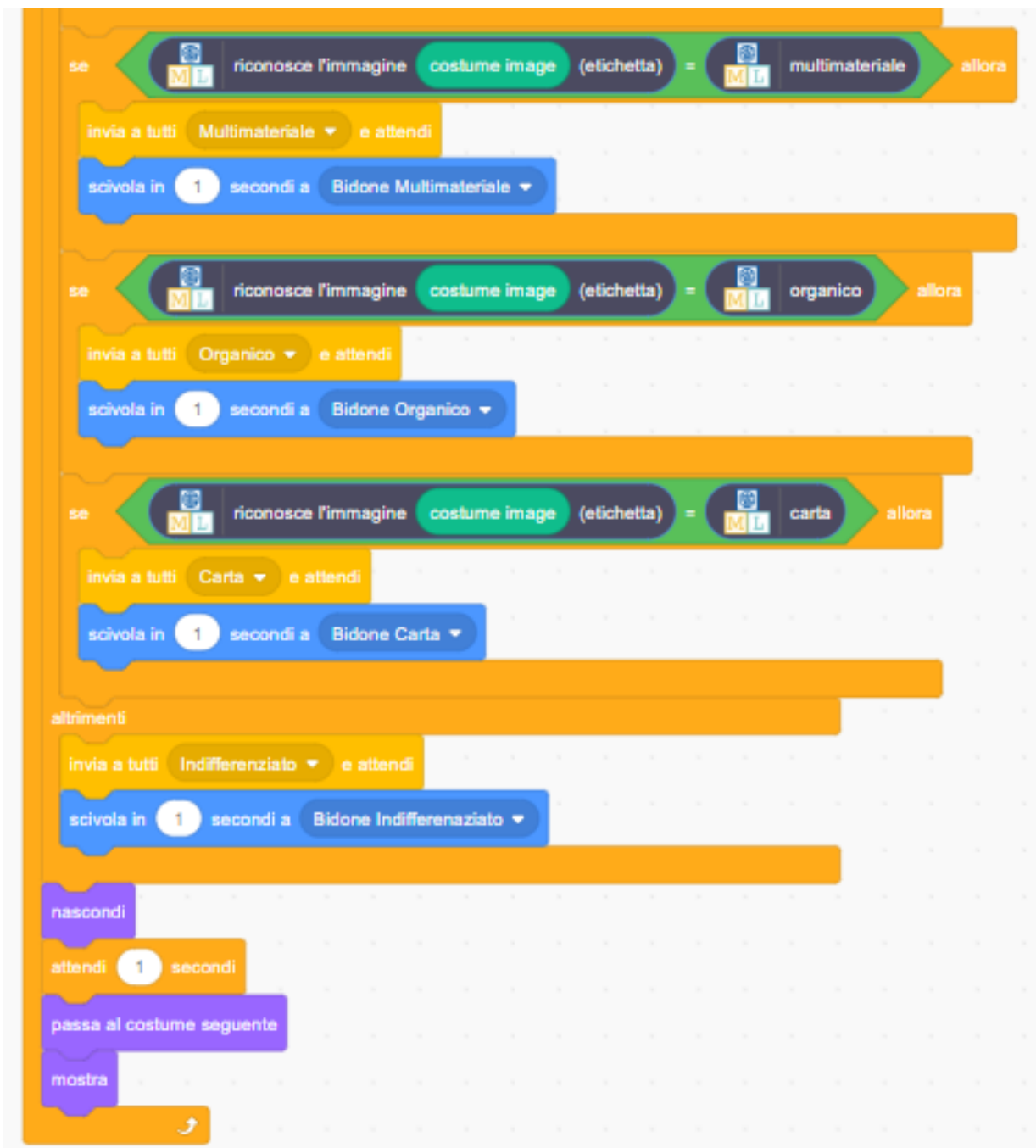
Settimo passo: aggiungere i mastelli

Per aumentare ulteriormente la difficoltà del gioco, e per renderlo ancora più interattivo coinvolgendo il giocatore in una vera e propria sfida, è possibile anche aggiungere i mastelli, e far sì che dopo che l'utente ha risposto al quiz, il rifiuto venga gettato nel mastello corretto.



Per fare questo, quindi, è necessario aggiungere cinque Sprite, uno per ogni mastello. E successivamente si andrà a modificare il codice dello sprite Rifiuti.

```
quando si clicca su [bandierina]
per sempre
  vai a x: -96 y: -77
  invia a tutti Domanda e attendi
  se riconosce l'immagine costume image (confidenza) > 50 allora
  se riconosce l'immagine costume image (etichetta) = vetro allora
    invia a tutti Vetro e attendi
    scivola in 1 secondi a Bidone Vetro
```



I Chatbot e gli Assistenti intelligenti.

Fondamentalmente, un **chatbot** è un **software che simula ed elabora le conversazioni umane** (scritte o parlate), consentendo agli utenti di interagire con i dispositivi digitali come se stessero comunicando con una persona reale.

In questa esperienza esamineremo un uso domestico comune del machine learning: assistenti intelligenti come **Siri, Alexa, Microsoft Cortana, Samsung Bixby o Google Home** che possono svolgere semplici lavori per te quando lo chiedi, come impostare una sveglia, avviare un timer o riprodurre un po' di musica. .

Gli assistenti intelligenti sono sistemi ML addestrati a riconoscere il significato del testo. Hai visto che puoi addestrare un computer in modo che quando gli dai un po' di scrittura, il computer possa riconoscere ciò che stai cercando di dire. E se un computer può capire cosa intendi, può capire anche cosa gli stai chiedendo di fare.



creare un programma che categorizza il testo in base al riconoscimento dell'intenzione del testo (classificazione dell'intento), raccogliamo un gran numero di esempi di ogni tipo di comando che vogliamo che riconosca e quindi utilizziamo il ML per addestrare un modello.

La **classificazione degli intenti è utile per costruire sistemi informatici** con cui possiamo interagire in modo naturale. Ad esempio, un computer potrebbe riconoscere che quando dici “Accendi la luce”, l’intenzione è che la luce venga accesa. Questa è descritta come **un'interfaccia in linguaggio naturale**. In altre parole, invece di dover premere un interruttore per accendere la luce, si utilizza il linguaggio naturale – un linguaggio che si è evoluto naturalmente negli esseri umani, non progettato per i computer – per comunicare tale intento

Il **computer impara dagli schemi negli esempi che gli forniamo**: schemi nelle parole che scegliamo, nel modo in cui esprimiamo i comandi, come combiniamo le parole per certi tipi di comandi e quando usiamo comandi brevi anziché più lunghi, solo per citare alcuni.

Altre applicazioni di Chatbot

Ai fini di evitare confusioni, va fatto un inciso per i BOT, quali ChatGPT di OpenAI, Bing di Microsoft, Gemini di Google, Claude di Anthropic e altri. Questi sono strumenti di **intelligenza artificiale generativa**, che permettono di generare un contenuto, ma non sono oggetto di studio in questo momento.

I **chatbot sono stati anche integrati, nelle app di messaggistica**, quali Telegram o Whatsapp, dove particolari frasi attivano specifiche interazioni. Vi sono funzionalità estese di un chatbot su smartphone, che possono attivare un aspirapolvere, avviare una lavatrice o registrare un brano musicale.



Gli Assistenti intelligenti in Scratch.

Qui creeremo un assistente virtuale intelligente in grado di riconoscere i tuoi comandi ed eseguire le tue istruzioni. Un sistema domotico, in grado di ricevere comandi, nel linguaggio naturale, per ottimizzare i consumi quotidiani.



Per cominciare, si addestrerà il modello ML a riconoscere i comandi per accendere o spegnere due dispositivi, una ventola e una lampada.

Codificare il progetto senza machine learning

È **utile vedere la differenza che fa il machine learning**, provando prima a codificare un progetto di intelligenza artificiale senza di esso.

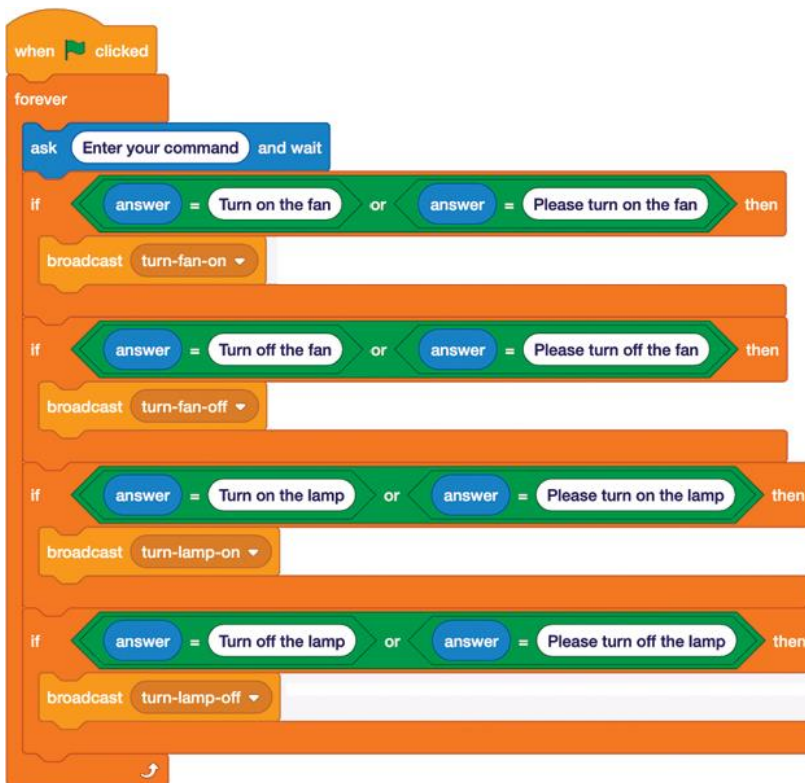
Puoi saltare questo passaggio se ritieni di avere una buona conoscenza della differenza tra un approccio basato su regole ed il machine learning e preferisci passare direttamente all'utilizzo del machine learning.

Dall'ambiente scratch 3 di MLK, si può accedere ai modelli di progetto, nella parte superiore dello schermo. I modelli di progetto includono progetti iniziali per risparmiare tempo, si può importare il modello "Aula Intelligente".

Questo script ti chiede di inserire un comando. Se digiti Accendi (o spegni) la ventola (o la lampada), Scratch riproduce l'animazione corrispondente, cambiando il costume. I due sprites dei dispositivi contengono ciascuno 2 costumi (acceso/spento), che sono intercambiati, quando la richiesta coincide con il testo del comando.

Lo **sfondo di background simula l'agente intelligente** che riconosce i comandi e li rilancia attraverso dei messaggi ai dispositivi che reagiscono variando il loro costume. Qui sotto la codifica di un assistente intelligente utilizzando le regole

Metti alla prova il tuo progetto facendo clic sulla bandiera verde.



Digita il comando Accendi la ventola e controlla che la ventola inizia davvero a girare.

Cosa succede se scrivi qualcosa di sbagliato? Cosa succede se si cambia la dicitura (ad esempio "Accendi il ventilatore, per favore")? Cosa succede se non menzioni la parola ventilatore (ad esempio: "Ho molto caldo, abbiamo bisogno di un po' d'aria qui!")?

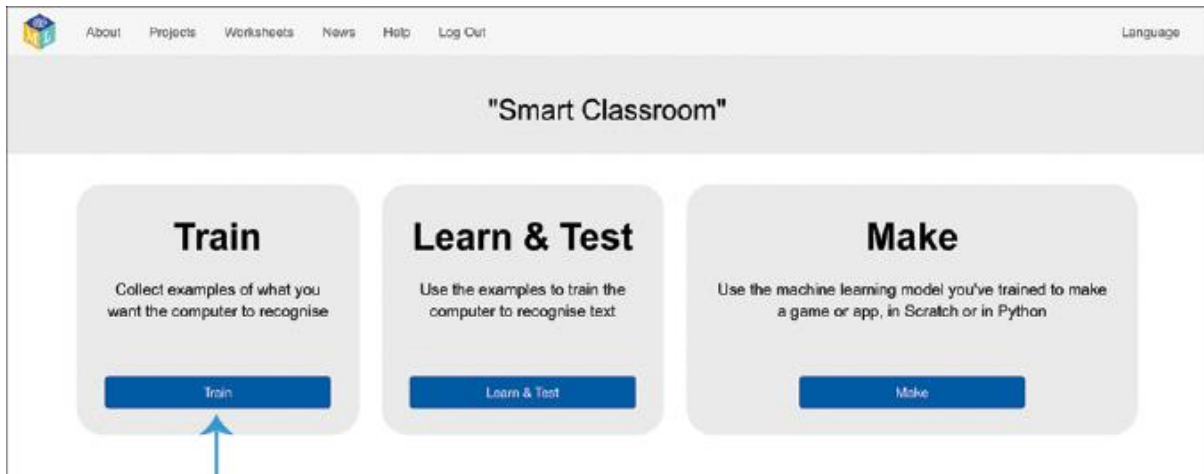
Perché non funzionano? **Pensi che sia possibile scrivere uno script che funzioni con qualsiasi frase di questi quattro comandi?** Saremo obbligati a scrivere un codice molto lungo dove verificare tutti i comandi. Possiamo pensare di

basarci sulla **gestione di 4 categorie di intenzioni** con il machine learning.

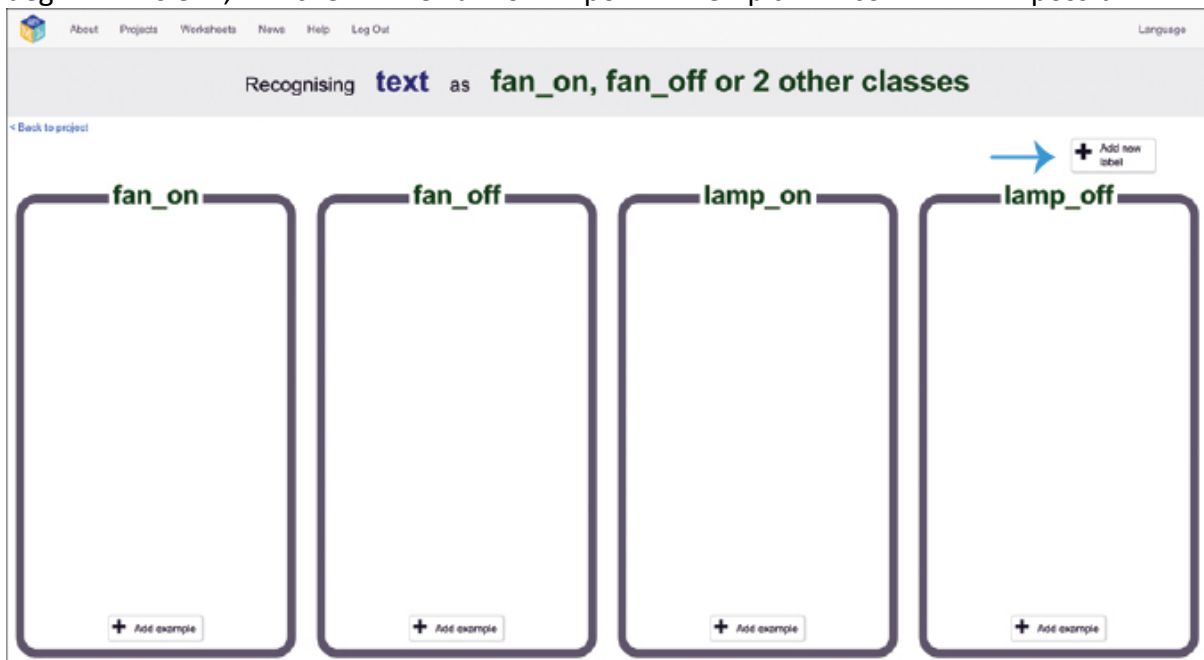
Qui **hai creato un progetto AI utilizzando un approccio basato su regole anziché ML**. Provando altre tecniche come questa e vedendo dove non sono all'altezza, si **può capire meglio perché il machine learning è preferito per così tanti progetti**. Si possono rendere i sistemi più complessi e generici nel rispondere a diversi stimoli.

[Addestrare un modello di comandi con il machine learning.](#)

Crea un nuovo progetto ML, chiamalo Smart Classroom e impostalo per imparare a riconoscere il testo nella tua lingua preferita. "Creazione di un nuovo progetto ML" nella sezione Progetti / Nuovo progetto.

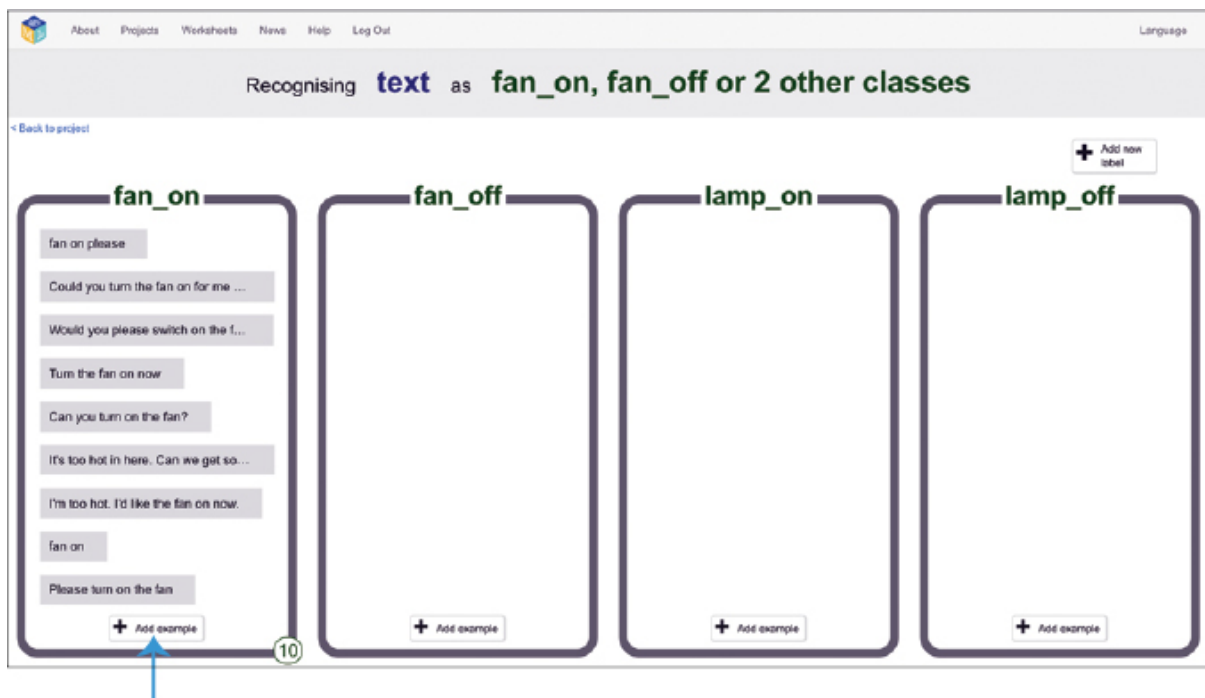


La prima fase consiste nel creare 4 bucket (cestini di raccolta) degli intenti. Questi saranno i nomi degli insiemi, che verranno poi riempiti con i possibili intenti.



Fare clic su **Aggiungi nuova etichetta**, creare un bucket di addestramento denominato *fan on*. Ripetere questo passaggio per creare altri tre contenitori di formazione denominati ventola spenta, lampada accesa e lampada spenta. (I caratteri di sottolineatura verranno aggiunti automaticamente.)

Fai clic su **Aggiungi esempio** nel bucket *fan_on* e digitare un esempio di come chiederesti a qualcuno di accendere la ventola.



Può essere breve (ad esempio, "accendi la ventola, per favore") o lungo ("Potresti accendermi la ventola adesso, per favore?").

Può essere educato ("Vuoi accendere il ventilatore, per favore?") o meno educato ("Accendi il ventilatore adesso").

Può includere le parole ventilatore e acceso ("Puoi accendere il ventilatore?") o nessuna delle due ("Fa troppo caldo qui. Possiamo far entrare un po' d'aria qui, per favore?").

Introduci quanti ne puoi immaginare, come mostrato nella Figura 10-6. Ti servono almeno cinque esempi, ma più grande è il nostro dataset, migliore sarà il risultato!

Fare clic su Aggiungi esempio nel bucket fan_off, come mostrato a seguito.

Questa volta, digita tutti gli esempi che ti vengono in mente per chiedere a qualcuno di spegnere la ventola. Hai bisogno di almeno cinque esempi.

Questi sono gli esempi che il tuo modello ML utilizzerà per apprendere l'aspetto di un comando "fan off".

Prova a includere alcuni esempi che non includano le parole fan o off.

Recognising **text** as **fan_on**, **fan_off** or 2 other classes

fan_on

fan_off

lamp_on

lamp_off

+ Add new label

+ Add example

+ Add example

+ Add example

+ Add example

Ripeti questo processo per gli ultimi due bucket, finché non avrai almeno cinque esempi per tutti e quattro i comandi, come mostrato nella Figura 10-8.

Recognising **text** as **fan_on**, **fan_off** or 2 other classes

fan_on

fan_off

lamp_on

lamp_off

+ Add new label

+ Add example

+ Add example

+ Add example

+ Add example

Fai clic su Torna al progetto nell'angolo in alto a sinistra dello schermo.

Fare clic su Impara e prova.

Fare clic su **Addestra nuovo modello di machine learning**, come mostrato nella figura seguente. Il computer utilizzerà gli esempi che hai scritto per imparare a riconoscere i tuoi quattro comandi. L'operazione potrebbe richiedere un minuto.

Machine learning models

[← Back to project](#)

What have you done?

You have collected examples of text for a computer to use to recognise when text is fan_on, fan_off or 2 other classes.

You've collected:

- 10 examples of fan_on,
- 10 examples of fan_off,
- 10 examples of lamp_on,
- 10 examples of lamp_off

What's next?

Ready to start the computer's training?

Click the button below to start training a machine learning model using the examples you have collected so far

(Or go back to the [Train](#) page if you want to collect some more examples first.)

Info from training computer:

[Train new machine learning model](#) ←

Dopo aver addestrato un modello ML, lo testiamo per vedere quanto è efficace nel riconoscere nuovi comandi. Digitare un comando nella casella Test.

Possiamo **valutare se il nostro comando è stato compreso a pieno**, o con quale precisione viene associato ad una delle precedenti categorie.

NOTA Assicurati di **testare il modello con comandi che non hai incluso** nei bucket di formazione.

NOTA Non ti interessa se il computer può ricordare ciò che gli hai già detto, ma se è in grado di riconoscere comandi che non ha mai visto prima.

Se il modello commette errori, puoi tornare alla fase Addestramento e aggiungere altri esempi dei comandi che continua a sbagliare. È come se un insegnante sfruttasse il risultato scarso dell'esame di uno studente per capire quali materie rivedere con lo studente per aiutarlo a migliorare la sua comprensione.

Dopo aver aggiunto altri esempi, torna alla fase di apprendimento e test e **addestra un nuovo modello ML**. Quindi provalo di nuovo per vedere se il computer è migliore nel riconoscere i comandi.

NOTA. Come si può osservare il sistema non è in grado di riconoscere comandi, su cui non è stato addestrato. Non è in grado di individuare comandi simili, se non esattamente corrispondenti, perché non era questo l'obiettivo del modello.

Nei sistemi di ML **l'insieme dei dati (dataset) di addestramento**, utili ad accendere / spegnere i dispositivi, **sono codificate / compresse** nelle diverse categorie, così da formare il **MODELLO di ML**. **Questo contiene le potenziali (possibili) risposte del sistema**, quando l'utente rilancia uno stimolo.

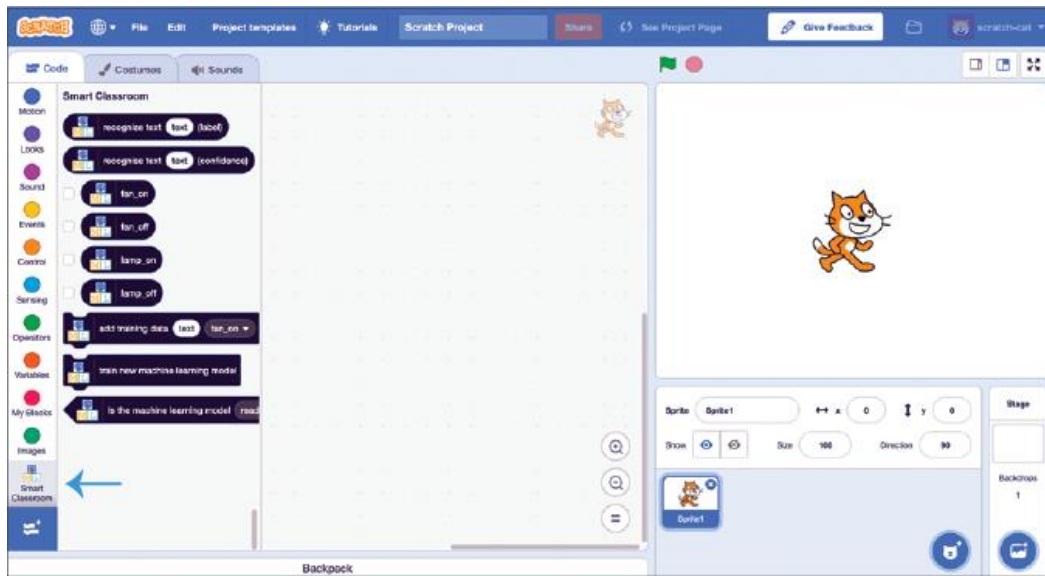
Codificare il tuo assistente intelligente con ML

Ora che disponi di un modello ML in grado di riconoscere i tuoi comandi, puoi ricreare il progetto precedente per utilizzare ML invece delle regole usate in precedenza.

Fai clic su Torna al progetto nell'angolo in alto a sinistra dello schermo.

Fare clic su Crea, salva il progetto come "Smart Classroom ML", per differenziarlo dal precedente.

Fare clic su Scratch 3, quindi su Apri in Scratch 3 per aprire una nuova finestra in Scratch. Dovresti vedere un nuovo set di blocchi per il tuo progetto ML nella casella degli strumenti.

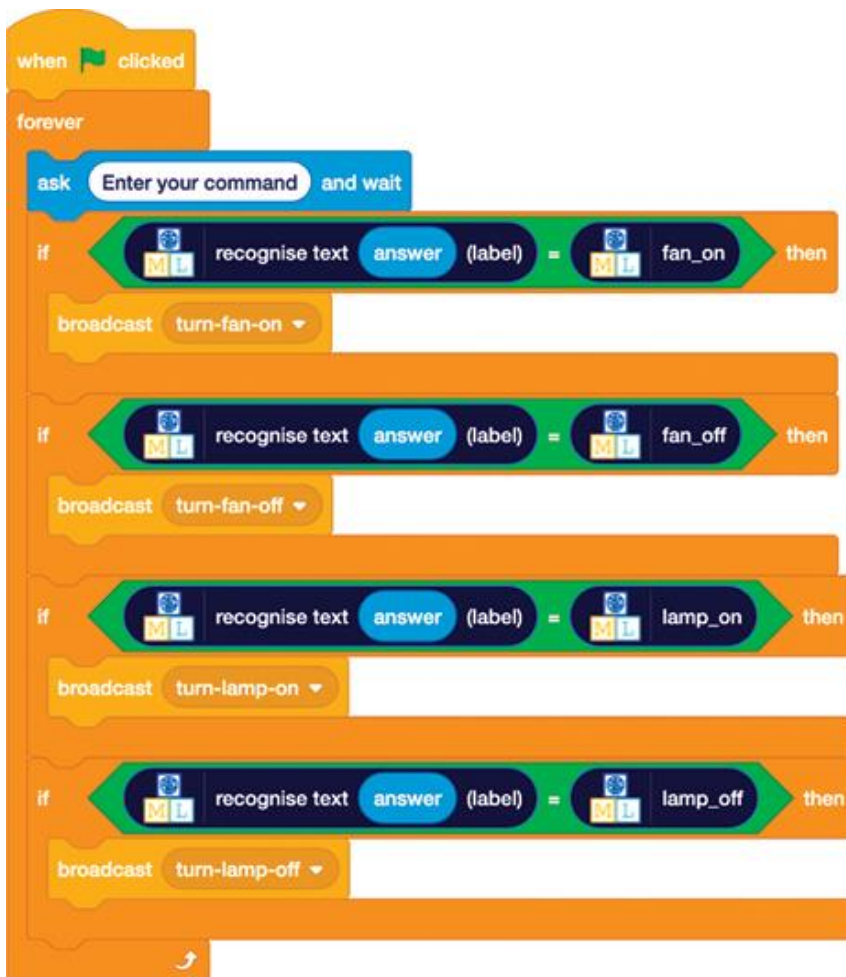


Fai clic su Modelli di progetto nella barra dei menu in alto e scegli il modello Smart Classroom. I modelli sono già preimpostati con gli elementi grafici utili.

Copiare lo script mostrato nella figura successiva..

Nell'approccio del ML, non si confronta direttamente il comando con l'azione da svolgere. In un primo momento attraverso il modello, si cerca di capire a quale categoria di istruzioni, appartenga il nostro intento.

NOTA. Quando fornisci comandi a questo script, questo utilizzerà il tuo modello ML per riconoscere il comando ed eseguire l'istruzione.



PROVA IL TUO PROGETTO

Metti alla prova il tuo progetto facendo clic sulla bandiera verde e inserendo una serie di comandi, formulati in molti modi diversi. **Scopri le prestazioni attuali del tuo assistente intelligente rispetto alla versione che non utilizzava il machine learning.**

RIVEDI E MIGLIORA IL TUO PROGETTO

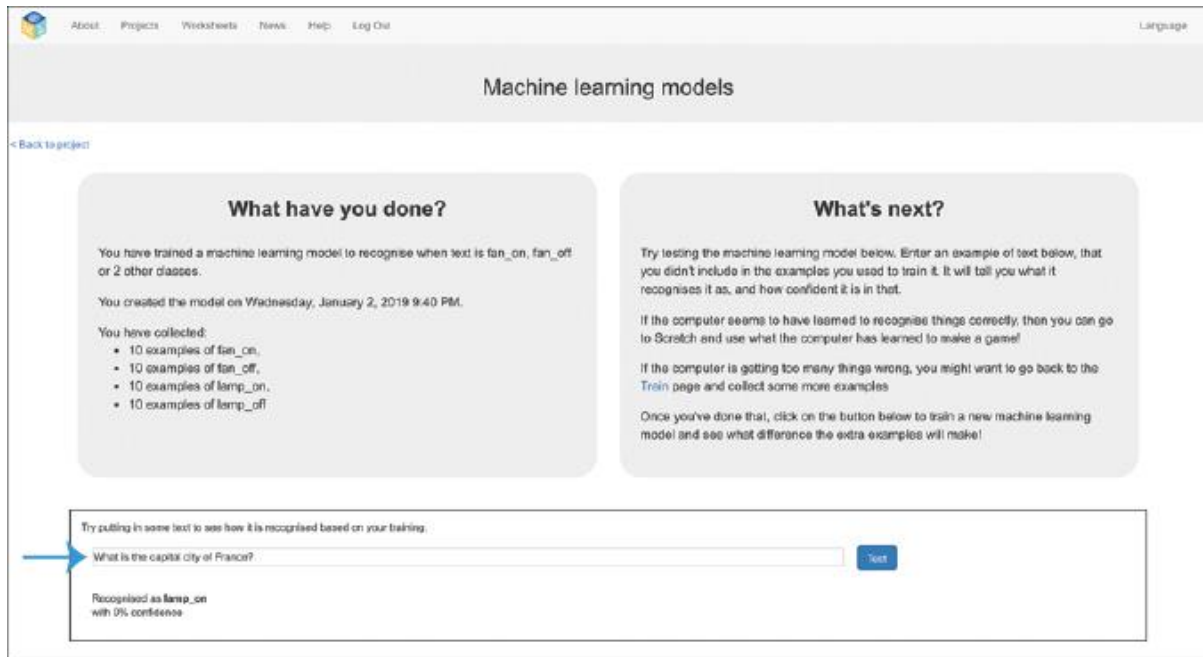
Hai creato il tuo assistente intelligente: una versione virtuale di Alexa di Amazon o Siri di Apple in grado di comprendere ed eseguire i tuoi comandi! **Cosa potresti fare per migliorare il modo in cui si comporta?**

Il punteggio di fiducia dell'assistente intelligente.

Nella fase di apprendimento e test, **dovresti aver notato il punteggio di confidenza** visualizzato quando hai testato il tuo modello. Questo indica quanto è sicuro che il computer abbia riconosciuto un comando.

Torna ora alla fase Impara e prova e prova a digitare qualcosa che non rientra in uno dei quattro comandi che il computer ha imparato a riconoscere.

Ad esempio, potresti provare "Qual è la capitale della Francia?" come mostrato nella Figura 10-13.



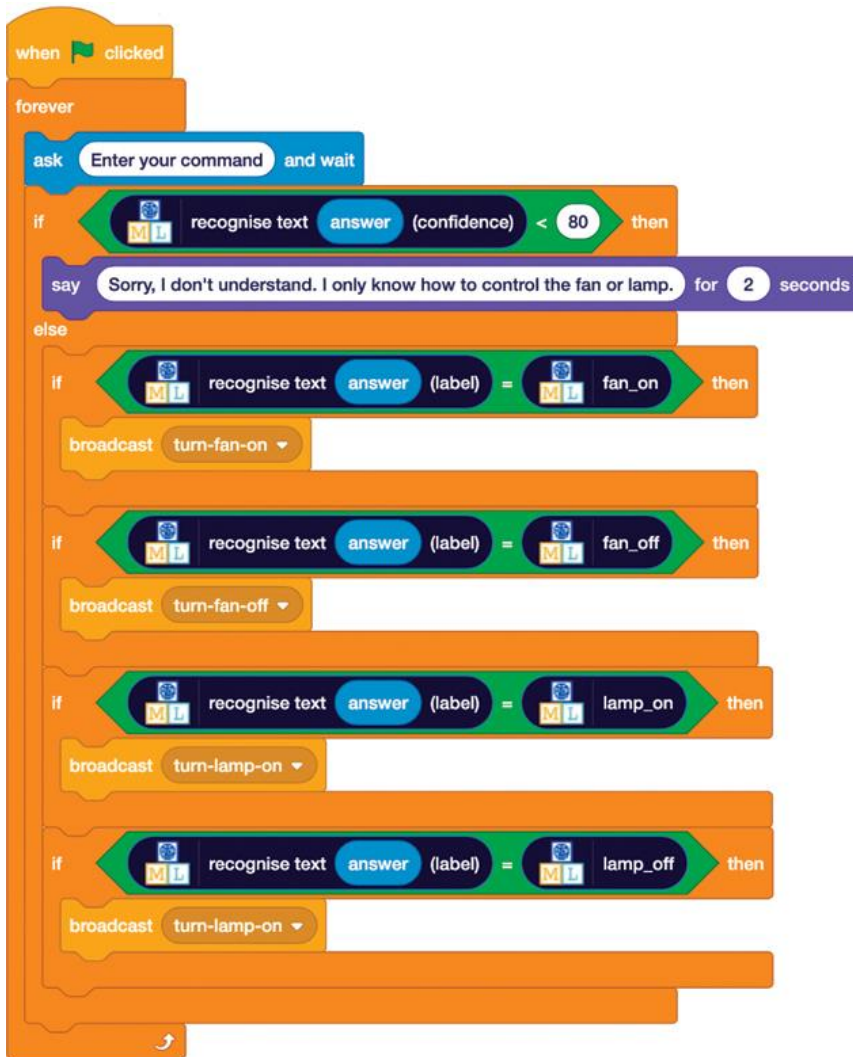
Il modello ML lo ha riconosciuto come "lampada accesa", ma aveva lo 0% di fiducia in quella classificazione. Questo era il modo in cui il modello ML mi diceva che non aveva riconosciuto il comando.

"Qual è la capitale della Francia?" non assomiglia a nessuno degli esempi che ho fornito al modello ML. **La domanda non corrisponde agli schemi identificati negli esempi che ho utilizzato per addestrarlo.** Ciò significa che non può riconoscere con sicurezza la domanda come uno dei quattro comandi che è stato addestrato a riconoscere.

Il tuo modello ML potrebbe avere un **livello di confidenza maggiore di zero, ma dovrebbe comunque essere un numero relativamente basso.** (In caso contrario, prova ad aggiungere altri esempi con cui addestrare il tuo modello ML.)

Sperimenta con altre domande e comandi che non hanno nulla a che fare con un ventilatore o una lampada. **Confronta i punteggi di affidabilità forniti dal tuo modello ML con quelli visualizzati quando riconosce i comandi effettivi di accensione, spegnimento ventola, accensione lampada e spegnimento lampada.** Che tipo di punteggi di confidenza fornisce il tuo modello ML quando riconosce correttamente qualcosa?

Qui sotto uno script per l'utilizzo dei punteggi di confidenza nel progetto ML



Una volta che hai un'idea di come funzionano i punteggi di confidenza per il tuo modello ML, puoi utilizzarlo nel tuo progetto Scratch. Aggiorna il tuo script in modo che assomigli alla Figura 10-14. Ora, **se il modello non è sicuro almeno all'80% di aver compreso correttamente** il comando, mostrerà una risposta "scusa" per 2 secondi e **non eseguirà l'azione**. Dovrai modificare il valore 80 in questo script in una percentuale che corrisponda al comportamento del tuo modello ML.

Tutti gli script, qui riportati, ad esclusione di quelli qui a fianco, sono i comportamenti del background bianco. Lo sfondo simula l'etere dove sarebbero diffusi vocali del chatbot, che per una



questione didattica, sono invece visualizzati come testi. Le interazioni con l'utente sono proposte con dei fumetti, dove viene dato l'output del chatbot e delle caselle di testo

dove vengono introdotti gli intenti da tastiera.

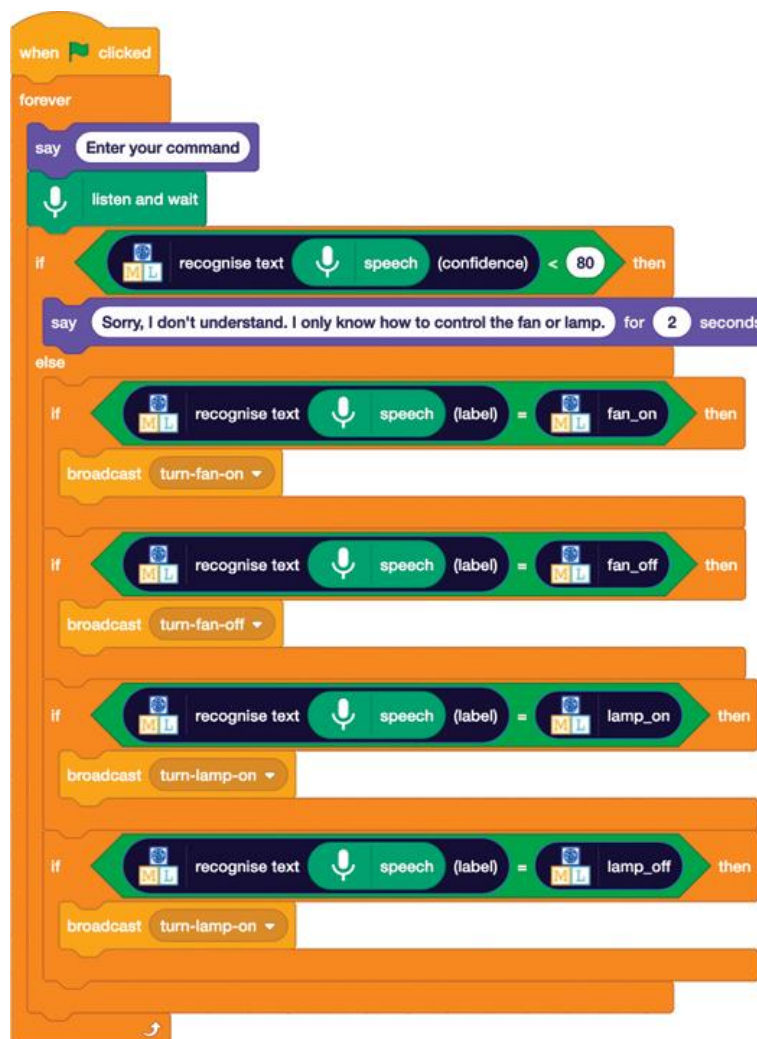
Qui le caratteristiche del ventilatore, che reagisce alla ricezione dei messaggi, in modo simile si possono realizzare quelli per la lampada.

Assistente intelligente con l'immissione vocale.

Si può modificare il progetto per renderlo più simile agli assistenti intelligenti del mondo reale utilizzando l'input vocale invece della digitazione.

Nella casella degli strumenti, fare clic sull'icona Libreria estensioni (assomiglia a due blocchi e un segno più), aggiungere l'estensione Speech to Text e aggiornare lo script come mostrato nella Figura 10-5.

NOTA Al momento attuale, l'estensione Scratch Speech to Text può essere utilizzata solo nel browser Web Google Chrome.



Raccolta dati per migliorare l'assistente intelligente.

Cos'altro potresti fare per migliorare il tuo progetto?

Il machine learning viene spesso utilizzato per riconoscere il testo perché è più veloce che dover scrivere regole. Ma **addestrare adeguatamente un modello richiede moltissimi esempi.** Per costruire questi sistemi nel mondo reale, avremmo bisogno di modi più efficienti per raccogliere esempi piuttosto che semplicemente digitarli tutti da soli come hai fatto finora. Ad esempio, invece di chiedere a una persona di scrivere 100 esempi, potrebbe essere meglio chiedere a 100 persone di scrivere un esempio ciascuna. O 1.000 persone. O 10.000 persone.

Se riesci a capire quando il tuo modello ML ha qualcosa di sbagliato, puoi raccogliere più esempi da aggiungere ai tuoi bucket di formazione. Ad esempio, **cosa succede se il modello ML ha un punteggio di confidenza molto basso?** O cosa succede se qualcuno continua a dare un comando simile in modi leggermente diversi? Ciò probabilmente significa che il modello ML non riconosce correttamente i comandi o non fa ciò che la persona desidera, e questo è un feedback utile per la tua formazione.

Esistono **molti modi per indovinare che qualcosa non ha funzionato bene.** E ogni volta che ciò accade, si tratta di un esempio che potresti raccogliere e aggiungere a uno dei tuoi bucket di formazione in modo che un modello ML più recente possa funzionare un po' meglio la prossima volta.

Utilizziamo tutti questi tipi di **tecniche (raccolta di esempi di formazione da un gran numero di persone, ottenere feedback dagli utenti e molto altro)** per aiutarci a costruire computer e dispositivi in grado di capire cosa intendi.

COSA HAI IMPARATO

In questo capitolo abbiamo esaminato come viene utilizzato il machine learning per riconoscere il significato del testo e come può essere utilizzato per creare sistemi informatici in grado di comprendere ciò che intendiamo e fare ciò che chiediamo.

Nel tuo progetto, hai utilizzato lo stesso tipo di tecnologia ML che abilita assistenti intelligenti come Alexa di Amazon, Google Home, Cortana di Microsoft e Siri di Apple. Le interfacce in linguaggio naturale ci consentono di dire ai nostri dispositivi cosa vogliamo che facciano utilizzando lingue come l'inglese, anziché solo premendo schermi o pulsanti.

Quando chiedi a uno smartphone che ore sono, o di impostare una sveglia o un timer, o di riprodurre la tua canzone preferita, il computer deve classificare quel comando. È necessario prendere quella serie di parole che hai scelto e riconoscere il loro intento.

I creatori di smartphone e assistenti intelligenti hanno addestrato un modello ML a riconoscere il significato dei comandi dell'utente elaborando un elenco di categorie: tutti i possibili comandi che pensavano gli utenti potessero voler impartire. E poi per ognuno hanno raccolto tantissimi esempi di come qualcuno potrebbe dare quel comando.

Sia in questo progetto che nel mondo reale, **il processo di apprendimento funziona in questo modo:** Prevedi i comandi che potresti dare.

Raccogli esempi di ciascuno di questi comandi.

Utilizza questi esempi per addestrare un modello ML.

Scrivi o codifica ciò che vuoi che il computer faccia quando riconosce ciascun comando.

Per creare un vero assistente intelligente, dovresti ripetere questi passaggi per migliaia di comandi, non solo quattro. E avresti bisogno di migliaia, o decine di migliaia, di esempi per ogni comando.













ESTENSIONE DEL PROGETTO.

Prova ad estendere il tuo progetto, controllando un dispositivo aggiuntivo, quale l'apertura/chiusura della porta di un frigo.. Introduci un nuovo sprite del frigorifero, con 2 costumi ed i suoi 2 bucket per contenere i relativi comandi.

<p>NOTA Lo sapevi che riducendo i tempi in cui il frigorifero rimane aperto possiamo ridurre i consumi energetici, conseguentemente la nostra impronta ambientale. Se diminuiamo il tempo di apertura, impediamo al calore di accedere alla cella interna del frigorifero. In questo modo il compressore non dovrà attivare un nuovo ciclo di raffreddamento.</p>
--


Glossario dei Blocchi

Qui di seguito è possibile visionare tutti i blocchi e le loro funzionalità

Blocchi Movimento	
BLOCCO	DESCRIZIONE
	Muove lo sprite avanti (usando valori positivi) o indietro (usando valori negativi)
	Ruota lo sprite in senso orario per i gradi indicati
	Ruota lo sprite in senso antiorario per i gradi indicati
	Fa spostare lo sprite in determinata direzione (0=su, 90=destra, -90= sinistra, 180=giù)
	Fa puntare lo sprite verso il puntatore del mouse o verso un altro sprite
	Sposta lo sprite alla direzione x e y indicata
	Fa spostare lo sprite dove si trova il puntatore del mouse o un altro sprite
	Fa spostare lo sprite nel punto indicato dalle x e dalle y facendolo scivolare nel tempo indicato.
	Modifica la posizione X dello sprite del valore specificato
	Porta lo sprite dove X è al valore specificato
	Modifica la posizione Y dello sprite del valore specificato
	Porta lo sprite dove Y è al valore specificato

rimbalza quando tocchi il bordo	Fa ruotare lo sprite in direzione opposta quando tocca il bordo
posizione x	Riporta la posizione X dello sprite
posizione y	Riporta la posizione Y dello sprite
direzione	Riporta la direzione dello sprite (0=su, 90=destra, -90= sinistra, 180=giù)

Blocchi Aspetto

BLOCCO	DESCRIZIONE
passa al costume <input type="text" value="costume2"/>	Cambia l'aspetto dello sprite passando al costume specificato
passa al costume seguente	Cambia l'aspetto dello sprite passando al costume successivo della lista dei costumi (se il costume è l'ultimo della lista, ritorna al primo)
numero costume	Riporta il numero attuale del costume dello sprite
dire <input type="text" value="Ciao!"/> per <input type="text" value="2"/> secondi	Mostra il messaggio specificato in un fumetto (effetto parola) per il tempo indicato 
dire <input type="text" value="Ciao!"/>	Mostra un il testo specificato in un fumetto (effetto parola)
pensa <input type="text" value="Hmm..."/> per <input type="text" value="2"/> secondi	Mostra il messaggio specificato in un fumetto (effetto pensiero) per il tempo indicato
	Mostra un il testo specificato in un fumetto (effetto pensiero)
	Cambia l'effetto grafico dello sprite del valore specificato (selezionare dal menù il tipo di effetto grafico desiderato)

	Porta un effetto grafico al valore specificato (la maggior parte degli effetti grafici ha valore tra 0 e 100)
	Rimuove tutti gli effetti grafici dello sprite
	Cambia la dimensione dello sprite della quantità specificata
	Porta la dimensione dello sprite alla percentuale specificata (il valore fa riferimento alla dimensione originale dello sprite)
	Riporta la percentuale dell'attuale dimensione dello sprite
	Fa comparire lo sprite nello stage
	Fa scomparire lo sprite dallo stage
	Porta in primo piano lo sprite rispetto a quelli presenti sullo stage
	Muovi lo sprite indietro del numero di livelli indicato
	Cambia l'aspetto dello stage passando allo sfondo specificato
	Cambia l'aspetto dello stage passando allo sfondo successivo
	Riporta il numero dello sfondo attuale dello stage

Blocchi suono

BLOCCO	DESCRIZIONE
	Inizia a riprodurre il suono selezionato tramite il menu e passa SUBITO all'esecuzione del blocco successivo, anche se il suono non è terminato

	Inizia a riprodurre il suono selezionato tramite il menu e passa ATTENDE IL TERMINE del suono per l'esecuzione del blocco successivo
	Arresta la riproduzione di tutti i suoni
	Riproduce il suono di uno strumento a percussione, selezionabile dal menu, per il numero di battute specificato
	Suona una nota musicale per il numero di battute indicato (occorre utilizzare la tastiera per selezionare una nota)
	Fa una pausa del numero di battute indicato
	Passa allo strumento specificato. Lo strumento verrà utilizzato dallo sprite per i successivi blocchi "suona nota"
	Cambia il volume dei suoni dello sprite del valore indicato (tra 0 e 100, valori positivi e negativi)
	Porta al valore specificato il volume de suoni riprodotti dallo sprite
	Riporta il volume dei suoni riprodotti dallo sprite
	Cambia la velocità dei suoni riprodotti per il valore specificato
	Porta il tempo dei suoni riprodotti dallo sprite al valore di battute al minuto specificato
	Riporta il tempo dei suoni riprodotti dallo sprite (i valori sono in battute per minuto)

Blocchi Controllo

BLOCCO	DESCRIZIONE
--------	-------------

	Esegue lo script ad esso collegato quando si clicca sulla bandiera verde
	Esegue lo script associato quando viene premuto un tasto specificato
	Esegue lo script ad esso collegato quando si clicca su questo sprite
	Attende il numero di secondi indicato prima di riprodurre lo sprite successivo
	Esegue all'infinito i blocchi inseriti al suo interno
	Ripete il numero di volte indicato i blocchi inseriti al suo interno
	Invia un messaggio a tutti gli sprite (incluso se stesso) attivando l'attivazione di alcuni blocchi. Non attende l'esecuzione di tali blocchi prima di proseguire
	Invia un messaggio a tutti gli sprite (incluso se stesso) attivando l'attivazione di alcuni blocchi. Attende il termine dell'esecuzione di tali blocchi prima di proseguire.
	Esegue lo script ad esso collegato quando riceve un determinato messaggio
	Esegue un continuo controllo ed esegue i blocchi inseriti al suo interno quando si verifica la specifica condizione indicata.
	Se si verifica una determinata condizione indicata esegue i blocchi contenuti al suo interno
	Se si verifica una determinata condizione esegue i blocchi inseriti nella parte superiore, altrimenti, quando tale condizione non si verifica, quelli della parte inferiore

	Attende fino a quando la condizione indicata non risulta vera e solo allora esegue gli script associati
	Ripete i blocchi inseriti al suo interno fino a quando la condizione continua ad essere vera
	Ferma l'esecuzione dei blocchi ad esso collegati
	Ferma l'esecuzione di tutti i blocchi del progetto

Blocchi penna	
BLOCCO	DESCRIZIONE
	Cancella dallo stage tutto ciò che è stato disegnato con la penna e tutti i timbri
	Appoggia la penna sullo stage: lo sprite mentre si sposta traccia una linea
	Alza la penna dallo stage: lo sprite può spostarsi senza tracciare alcuna linea
	Cambia il colore della penna. Il colore viene scelto con il contagocce
	Usa la penna del colore indicato (0 = colore rosso; 100 = colore blu. Le tonalità rispecchiano i colori dell'arcobaleno)
	Cambia il valore indicato della luminosità della penna dello sprite
	Porta la luminosità della penna al valore specificato (0= molto scuro; 100=molto chiaro)
	Modifica la dimensione del tratto della penna
	Sa una penna delle dimensioni indicate
	Riproduce l'immagine dello sprite sullo stage

Blocco sensori	
BLOCCO	DESCRIZIONE
	La condizione è vera se lo sprite sta toccando il puntatore del mouse, un altro sprite o il bordo dello stage.
	La condizione è vera se lo sprite sta toccando il colore indicato (il colore viene scelto utilizzando il contagocce)
	La condizione è vera se il primo colore specificato sta toccando il secondo colore specificato (entrambi si scelgono con il contagocce)
	Visualizza la domanda indicata e una casella di testo in cui è possibile digitare una risposta. La conferma della risposta avviene dopo che è stato premuto il tasto "invio".
	Memorizza l'ultima risposta data tramite il blocco "chiedi e...attendi". Il valore è globale perché condiviso con tutti gli sprite.
	Riporta la posizione x del puntatore del mouse
	Riporta la posizione y del puntatore del mouse
	La condizione è vera quando il tasto del mouse viene premuto
	La condizione è vera quando il tasto specificato viene premuto
	Riporta la distanza dello sprite dal puntatore del mouse o da uno specifico sprite
	Riporta il valore, espresso in secondi, del cronometro. Il cronometro è sempre in esecuzione.

	Riporta il valore (espresso in secondi) del cronometro. Il cronometro è sempre in esecuzione.
	Riporta il valore relativo ad una specifica proprietà (posizione x, posizione y, direzione, numero, costume, dimensione e volume) di un determinato sprite
	Riporta il volume dei suoni catturati dal microfono del computer (il valore è compreso tra 0 e 100)
	La condizione risulta vera quando i suoni catturati dal microfono del computer superano il valore 30
	Riporta il valore del sensore specificato.
	La condizione è vera quando il pulsante è premuto o i contatti della Scratch Sensor Board sono connessi.

Blocchi Operatori

BLOCCO	DESCRIZIONE
	Esegue un'addizione e genera un risultato
	Esegue una sottrazione e genera un risultato
	Esegue una moltiplicazione e genera un risultato
	Esegue una divisione e genera un risultato
	Sceglie un numero intero a caso compreso nell'intervallo specificato
	La condizione risulta vera quando il primo valore è inferiore al secondo

	La condizione risulta vera quando il primo valore è uguale al secondo
	La condizione risulta vera quando il primo valore è superiore al secondo
	La condizione è vera quando entrambe le condizioni sono vere
	La condizione è vera quando una delle due condizioni è vera
	Inverte il valore di una condizione da vero in falso e viceversa
	Unisce due stringhe alfanumeriche o valori
	Riporta la lettera che occupa la posizione specificata di una stringa o di un valore
	Riporta il numero di caratteri di una stringa alfanumerica o di un valore
	Riporta il resto della divisione dei numeri specificati
	Riporta il numero intero più vicino al numero 0 al valore specificato
	Riporta il risultato della funzione selezionata (abs, sqtr, sin, cos, tan, asin, acos, atan, ln, log, e [^] ,10 [^]) applicata al numero specificato

Blocchi Variabili

BLOCCO	DESCRIZIONE
	Riporta il valore della variabile
	Porta il valore della variabile selezionata al valore indicato
	Modifica il valore della variabile significata della quantità specificata
	Apri un visualizzatore indicante il valore della variabile sullo stage

	Nasconde il visualizzatore
	Riporta tutti gli elementi contenuti nella lista
	Aggiunge alla lista l'elemento indicato. L'elemento viene inserito sempre in ultima posizione
	Rimuove un elemento occupante una specifica posizione nella lista, il primo, l'ultimo o tutti gli elementi
	Inserisce un elemento specifico nella posizione indicata (specifica, prima, ultima o qualunque) in una determinata lista
	Sostituisce un elemento occupante una determinata posizione (specifica, prima, ultima o qualunque), in una specifica lista con l'elemento indicato
	Riporta l'elemento occupante una determinata posizione (specifica, prima, ultima o qualunque), nella lista selezionata
	Riporta il numero totale degli elementi presenti nella lista selezionata
	La condizione risulta vera quando la lista contiene esattamente il valore specificato

Bibliografia:

<https://machinelearningforkids.co.uk/>

<https://scratch.mit.edu/>