



Relazione Gruppo 2

Documento tecnico di progetto.

GRUPPO 02
0612707416 - Gioia Iannuzzi
0612708273 - Luigi Montonetti
0612707862 - Arianna Paletta
0612708294 - Debora Villano

Indice

1	Obiettivo	2
2	Approccio al sistema di guida autonoma	2
2.1	Raccolta e struttura del dataset	2
2.2	Strategie iniziali di ottimizzazione	2
2.3	Struttura del sistema e algoritmo di predizione	3
2.4	Valutazione delle prestazioni	3
3	Ottimizzazione del Feature Set	3
3.1	Analisi dei target con PCA	4
4	Valutazione e Risultati	5
5	Conclusioni	5

1 Obiettivo

L'obiettivo del progetto **Gran Premio MIVIA 2025** è stato quello di sviluppare un sistema di guida autonoma capace di completare un giro in pista nel minor tempo possibile, senza la presenza di altri concorrenti. Il sistema si basa sull'algoritmo **K-Nearest Neighbors (KNN)** e su un **dataset** di training generato manualmente. Attraverso l'approccio **behavioral cloning**, il veicolo apprende il comportamento di guida umano, riproducendolo per seguire il tracciato in modo ottimale.

Per le simulazioni abbiamo utilizzato la macchina **Lotus Elise GT1 1997** e la pista **Forza** di A. Summer.



2 Approccio al sistema di guida autonoma

2.1 Raccolta e struttura del dataset

Il processo di raccolta dati è affidato al metodo `control(...)`, che, quando la registrazione è attiva, scrive su file i valori dei sensori e i comandi impartiti con periodicità di **50ms**:

1. **Attivazione/disattivazione della registrazione**
 - Tasto 1 → recording = true: inizia la scrittura sul file **dataset_50ms.csv**.
 - Tasto 0 → recording = false: termina la registrazione.
2. **Preparazione del file CSV** Alla prima scrittura, se il file non esiste o è vuoto, viene creata l'intestazione delle colonne.
3. **Raccolta e formattazione dei dati** In ogni intervallo di salvataggio vengono estratti i sensori
4. **Scrittura su file** Utilizzando BufferedWriter (in modalità append), il programma apre dataset.csv, aggiunge l'intestazione se appena creato, quindi scrive la riga formattata. Infine chiude BufferedWriter per garantire il salvataggio sicuro dei dati.

2.2 Strategie iniziali di ottimizzazione

Durante le fasi iniziali del progetto sono state esplorate diverse strategie per migliorare la qualità del dataset e affinare le previsioni del classificatore *K-Nearest Neighbors* (KNN). Le principali idee emerse sono state le seguenti:

1. **Filtraggio delle informazioni** Sono stati esclusi dal dataset i dati raccolti in condizioni sfavorevoli, come auto danneggiata, uscita di pista o velocità minima. Questa

selezione ha permesso di ridurre la presenza di dati non significativi, migliorando la precisione del classificatore.

2. **Generazione guidata del dataset** Durante la raccolta manuale dei dati, sono stati applicati criteri specifici per ottimizzare il comportamento del veicolo: limitazione dell'accelerazione in curva, massima velocità nei rettilinei e sterzata controllata in funzione della posizione rispetto alla pista.

L'adozione di queste strategie ha contribuito in modo significativo a migliorare la qualità delle previsioni, garantendo un sistema di guida autonoma più stabile ed efficiente.

Abbiamo anche valutato l'effetto della velocità massima:

- Aumentarla migliorava i tempi ma peggiorava la stabilità;
- Ridurla migliorava la stabilità ma rallentava troppo.

Abbiamo quindi adottato una strategia dinamica: mantenere alta la velocità nei rettilinei e ridurla automaticamente in curva per un buon compromesso tra prestazioni e affidabilità.

2.3 Struttura del sistema e algoritmo di predizione

Per la realizzazione del sistema di guida autonoma, i dati di addestramento vengono archiviati all'interno di una struttura KDTree, progettata per facilitare la ricerca efficiente dei punti più vicini in uno spazio multidimensionale. Durante la predizione, il test point viene normalizzato per garantire che ogni feature contribuisca equamente alla distanza euclidea.

Il metodo `kNearestNeighbors(testPoint, k)` restituisce i k campioni più simili. A partire da questi si calcola la media dei target continui (accelerazione, frenata, sterzata),

Durante la fase di progettazione è stata condotta un'analisi sperimentale sul parametro k , fondamentale per il funzionamento del classificatore K-Nearest Neighbors. Sono stati testati diversi valori di k valutando l'impatto sull'accuratezza delle predizioni e sulla stabilità del comportamento in pista. Il valore $k = 3$ è risultato essere il miglior compromesso: $k = 1$ mostrava un comportamento troppo reattivo e instabile, sensibile al rumore nei dati; valori di k più alti, rendevano il veicolo più conservativo ma anche meno preciso in situazioni dinamiche. La scelta di $k = 3$ ha quindi garantito una buona reattività mantenendo una certa robustezza alle variazioni del tracciato.

2.4 Valutazione delle prestazioni

Il dataset viene suddiviso in training set (80%) e test set (20%). Per ogni elemento del test set, si calcolano le azioni previste e si confrontano con i valori reali, ottenendo metriche quali l'errore quadratico medio e l'accuratezza discreta (tramite discretizzazione dei target in livelli). I risultati vengono esportati su file e analizzati al termine dell'esecuzione.

3 Ottimizzazione del Feature Set

Inizialmente sono state utilizzate tutte le feature disponibili per ottenere una visione completa dei dati. Tuttavia, questa scelta ha portato a un degrado delle prestazioni computazionali a causa dell'elevato numero di confronti richiesti dal classificatore KNN.

Per ovviare al problema, è stato selezionato un sottoinsieme di feature più rappresentative, tramite:

- analisi statistica (varianza e ridondanza tra feature),
- valutazione empirica con rimozione progressiva e misurazione dell'impatto su accuratezza e tempo di risposta.

Le feature finali selezionate sono: **Track2, Track5, Track8, Track9, Track10, Track13, Track16, TrackPosition, AngleToTrackAxis, Speed, SpeedY**.

Tutti i dati sono stati normalizzati. In particolare:

- Le **feature** (come velocità, posizione, sensori) vengono divise per un valore massimo noto, oppure trasformate in base alla loro scala naturale (ad esempio, angoli tra $-\pi$ e π).
- I **target** (accelerazione, frenata, sterzo) vengono normalizzati usando i minimi e massimi trovati nel dataset, secondo la formula:

$$\text{norm} = \frac{\text{valore} - \min}{\max - \min}$$

Questo processo garantisce che durante il *training* e la *predizione* i dati siano su una scala coerente, evitando che feature con valori numerici più grandi influenzino eccessivamente il calcolo della distanza nel KNN.

Infine, l'adozione del KDTree ha consentito un notevole miglioramento delle performance rispetto alla ricerca esaustiva, permettendo una gestione efficiente anche su dataset di grandi dimensioni.

3.1 Analisi dei target con PCA

I **target** sono stati **analizzati** tramite **riduzione dimensionale** con **PCA**. Le mappe risultanti illustrano come il sistema si comporta nello spazio delle feature in relazione ai target di regressione.

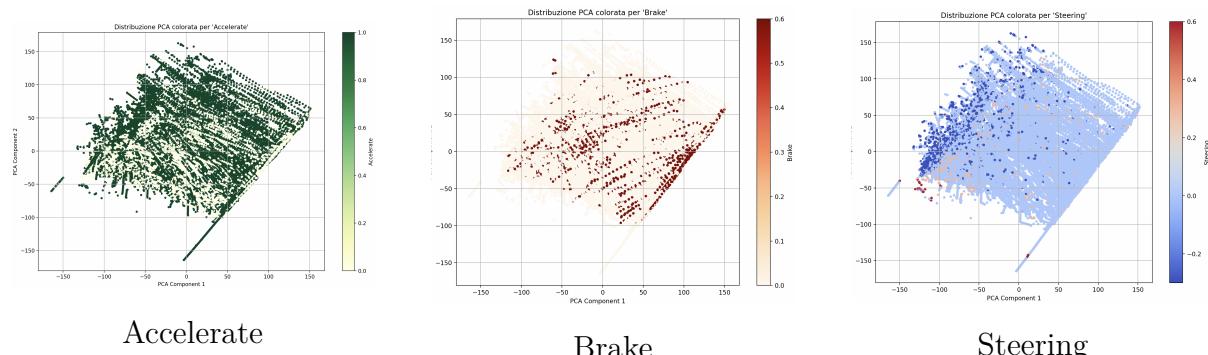


Figura 1: Visualizzazione PCA dei target: Accelerate, Brake, Steering

Si osserva una dominanza di **accelerazioni** elevate lungo l'intero tracciato, mentre le zone a bassa intensità sono concentrate in prossimità delle curve. La **frenata** si manifesta in cluster ben definiti, corrispondenti a momenti critici della guida; il comportamento appare

realistico e ben separato. Infine, la distribuzione delle **sterzate** mostra una simmetria tra destra e sinistra, indicando una buona copertura del comportamento di guida da parte del modello.

4 Valutazione e Risultati

Con la configurazione ottimizzata (11 feature + normalizzazione) e un $K = 3$, abbiamo effettuato 3 giri completi del circuito.

I risultati sono stati:

- **Giro migliore:** tempo migliore 2 minuti e 48 secondi;
- **Giro peggiore:** tempo peggiore 3 minuti e 12 secondi;

Rispetto alla versione con tutte le feature, il nuovo modello ha permesso di:

- Ridurre i tempi di risposta;
- Aumentare la stabilità in curva;
- Completare tutti i giri senza errori critici.

5 Conclusioni

Il progetto ha evidenziato l'efficacia dell'algoritmo KNN in un contesto di guida autonoma simulata, purché supportato da un'adeguata gestione del dato. La qualità delle predizioni è migliorata sensibilmente attraverso la selezione delle feature, la normalizzazione coerente e l'ottimizzazione dell'algoritmo di ricerca. Il sistema finale ha dimostrato buone prestazioni in termini di stabilità, tempi di risposta e affidabilità, completando più giri senza errori critici.