

# Predizione dell'Assegnazione degli Ordini Basata sui Dati Storici

Azar Bagheri  
Email: Azarbagheri45@gmail.com

**Sommario**—Questo progetto esplora un approccio basato sui dati per automatizzare l'assegnazione di operatori agli ordini in arrivo in un'azienda che ha registrato sistematicamente due anni di dati storici sugli ordini. Sono state applicate tecniche di machine learning per sviluppare modelli predittivi mirati all'identificazione di pattern ricorrenti nelle pratiche di assegnazione e alla previsione accurata delle future assegnazioni di operatori. Questo report fornisce una panoramica strutturata delle metodologie di pre-elaborazione dei dati, dei processi di feature engineering e dei modelli di machine learning valutati, tra cui reti neurali, alberi decisionali, foreste casuali, K-nearest neighbors, support vector machines e reti neurali ricorrenti. I risultati dimostrano prestazioni predittive moderate, con alcuni algoritmi che mostrano maggiore precisione nel catturare pattern complessi e non lineari nei dati.

## I. INTRODUZIONE

### A. Background

L'azienda gestisce manualmente l'assegnazione degli operatori, ruotando le assegnazioni in base alla disponibilità e ai turni di lavoro. Questo sistema è laborioso e non scalabile. Il progetto mira a sviluppare un modello basato sui dati per automatizzare le assegnazioni, migliorando l'efficienza e riducendo l'intervento umano.

### B. Obiettivo

L'obiettivo è analizzare i dati storici sugli ordini per prevedere l'operatore che si occuperà di nuovi ordini, basandosi sui pattern identificati nelle tendenze di assegnazione. Sono stati testati vari modelli di machine learning, sia algoritmi classici sia tecniche di deep learning, per identificare l'approccio predittivo più accurato.

## II. DATASET

### A. Panoramica dei Dati

Il dataset include due anni di dati storici sugli ordini contenenti:

- Un identificatore unico per ogni ordine.
- La marca temporale di completamento dell'ordine.
- L'ID dell'operatore che ha completato l'ordine.

### B. Qualità dei Dati e Sfide

Il dataset richiedeva pulizia per gestire i valori mancanti e standardizzare i formati temporali. Data la natura temporale delle assegnazioni, una sfida rilevante è stata identificare i pattern di rotazione nei turni degli operatori.

## III. ELABORAZIONE DEI DATI E PIPELINE

### A. Pulizia dei Dati

Sono stati adottati i seguenti passaggi:

- **Valori Mancanti:** Eliminazione delle righe con dati mancanti per garantire la coerenza.
- **Conversione Temporale:** Le marche temporali sono state convertite in formato datetime per estrarre ore, giorni e mesi.
- **Codifica Categoriale:** È stata applicata la codifica delle etichette per trasformare la variabile target (`operator_id`) da categorica a numerica.

### B. Feature Engineering

Sono state derivate diverse caratteristiche temporali e categoriali:

- **Caratteristiche Basate sul Tempo:** estratte ora, giorno, giorno della settimana e mese.
- **Carico di Ordini:** il numero di ordini per operatore per periodo è stato considerato per catturare l'efficienza dell'operatore.

Il dataset è stato poi suddiviso in set di addestramento e test (80% addestramento, 20% test) per la valutazione del modello.

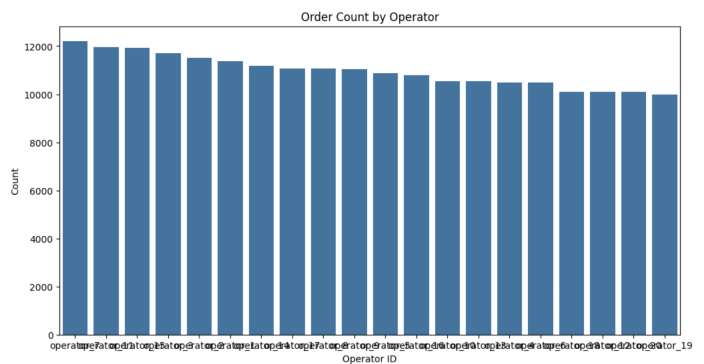


Figura 1. operator assignment patterns

## IV. APPROCCIO DI MODELLAZIONE

Tre script Python principali sono stati utilizzati per esplorare diversi modelli di machine learning.

### A. Script 1: Modelli Classici e Rete Neurale

#### Descrizione dei modelli:

- 1) **Rete Neurale**: rete completamente connessa con due layer nascosti (64 e 32 neuroni) con attivazione ReLU e layer di output softmax.
- 2) **Decision Tree Classifier**: modello non parametrico per catturare i confini decisionali.
- 3) **Random Forest Classifier**: modello ensemble con 100 stimatori per migliorare la generalizzazione.
- 4) **K-Nearest Neighbors (KNN)**: modello basato sulla distanza che utilizza i 5 vicini più prossimi.

La Random Forest ha ottenuto l'accuratezza più alta, superando leggermente Decision Tree e KNN.

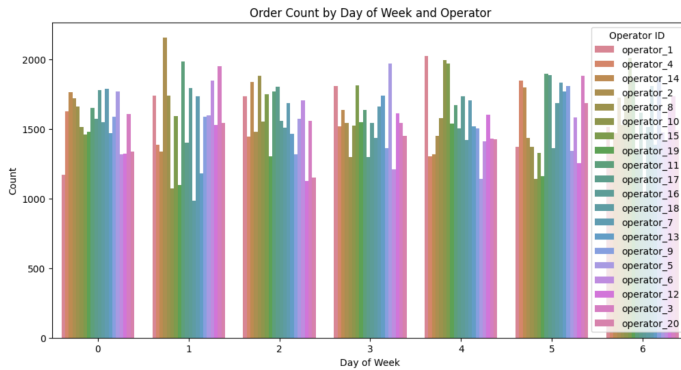


Figura 2. Explore the data

### B. Script 2: SVM e Rete Neurale Espansa

#### Descrizione dei modelli:

- 1) **Support Vector Machine (SVM)**: modello di classificazione con kernel RBF per gestire pattern non lineari.
- 2) **Rete Neurale**: rete neurale profonda con 32 e 16 neuroni nei layer nascosti, addestrata per 50 epoche.

### C. Script 3: Modello LSTM con Dati Temporal

#### Descrizione del modello:

- 1) **LSTM (Long Short-Term Memory)**: modello RNN per catturare pattern sequenziali. Utilizza due layer LSTM con attivazione Leaky ReLU, specificamente scelto per dati sequenziali.

Il modello LSTM ha raggiunto una precisione moderata, utile per gestire dati sequenziali ma limitato dalla complessità del dataset e richiedeva tempo di addestramento significativo.

## V. RISULTATI E DISCUSSIONE

### A. Confronto dei Modelli

- La Random Forest ha superato gli altri modelli classici.
- La precisione della rete neurale variava a seconda degli script, migliorando con epoche aumentate.

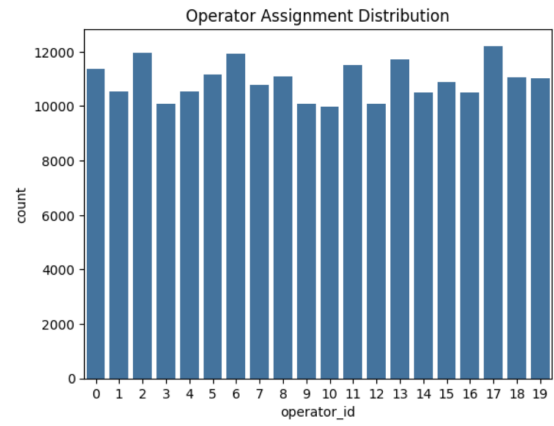


Figura 3. order

### B. Osservazioni Chiave

I pattern nel dataset, come le variazioni nell'assegnazione per giorno e carico di lavoro, sono stati visualizzati, suggerendo possibili influenze temporali.

### C. Sfide e Limitazioni

Una limitazione importante è stata la profondità sequenziale del dataset e la complessità dei turni operatori. Ulteriori dati, come orari dei turni, potrebbero migliorare la precisione delle previsioni.

## VI. CONCLUSIONI

Questo progetto ha dimostrato la fattibilità di prevedere le assegnazioni di operatori utilizzando dati storici e modelli di machine learning. La Random Forest ha mostrato maggiore potenzialità tra i modelli classici, mentre LSTM è stato utile per pattern sequenziali. Miglioramenti futuri potrebbero includere dati aggiuntivi, come turni degli operatori, e modelli sequenziali avanzati.