Specifiche tecniche del software – Predizione di carichi di potenza ospedalieri con Intelligenza Artificiale

Autori:

Andrea Bellome – Software Engineer, AI & Data Analysis

Pietro Iovino – Software Engineer, AI & Data Analysis

Leonardo Ratto – Software Engineer

**SCOPO**

Prevedere il consumo di energia elettrica degli ospedali/strutture sanitarie. In questo modo si riescono a prevedere con precisione i costi e ottimizzare l’allocazione dei fondi da parte del Ministero.

**DATI**

Il modello ha utilizzato per il training una serie temporale del consumo energetico di un’azienda (fonte: Università di Roma “La Sapienza”): il dataset contiene i dati giornalieri dal 1/1/2006 al 21/12/2020 dei carichi di potenza. Il dataset utilizzato è mostrato in Figura 1.

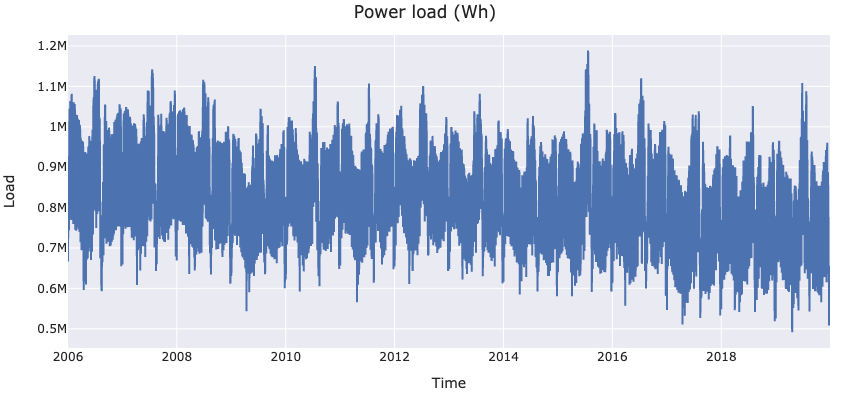


Figure 1. Carico di Potenza (Wh) per l’arco temporale 2006-2019

Una analisi di stagionalità è stata prodotta, e i risultati sono presentati nella Figura 2 e Figura 3. Come si vede, i carichi di potenza seguono una stagionalità annuale e settimanale, con minor carico nel fine settimana. Ad agosto, si osserva una generale diminuzione del carico. Questo è confermato anche dal box-plot prodotto in Figura 4, che mostra il carico per mese.

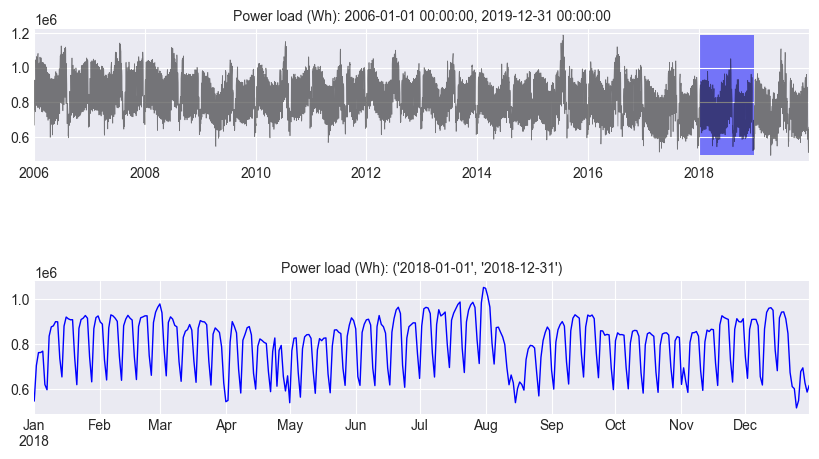


Figure 2. Zoom dei dati sul 2018.

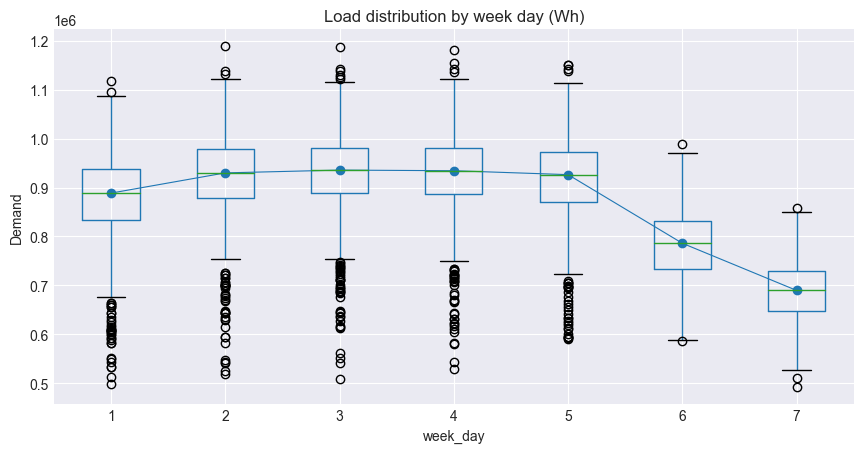


Figure 3. Box-plot del carico di potenza per giorno della settimana.

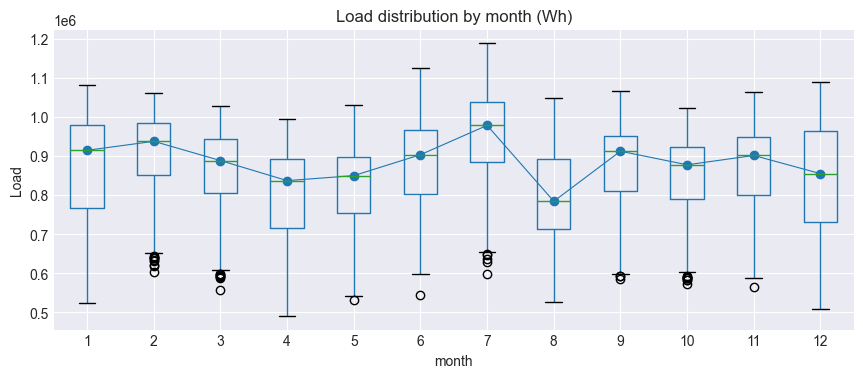


Figure 4. Box-plot del carico per mese.

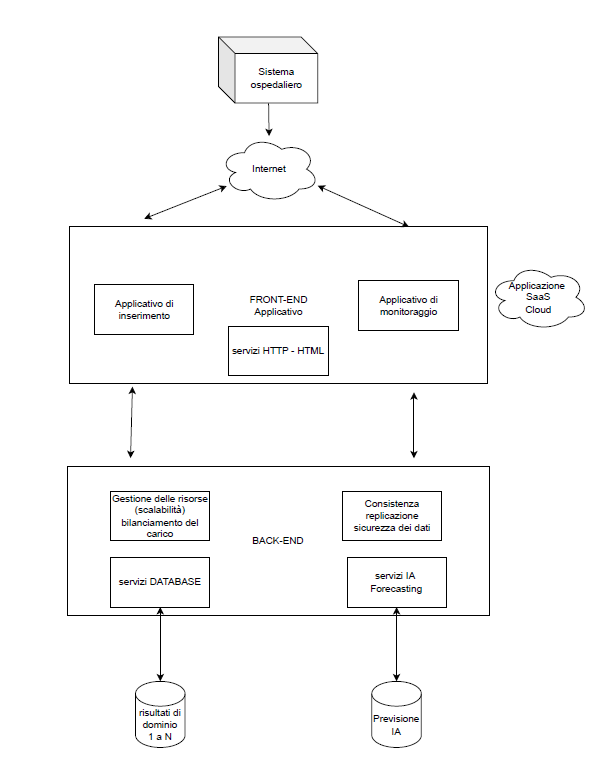
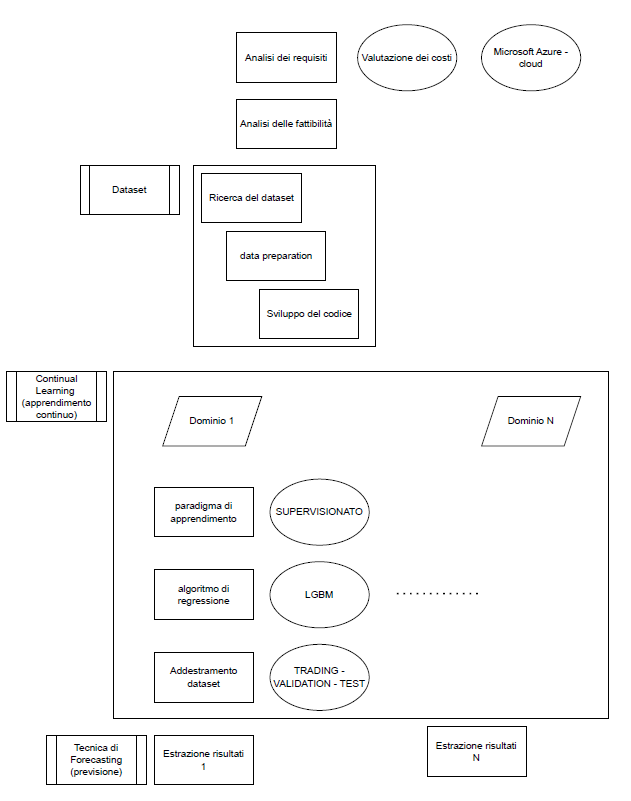
**SOLUZIONE**

Modello di machine learning: modello supervisionato con modello di auto-regressione LGBMRegressor, basato su un modello di ‘gradient-boosting’. Il modello può essere adattato a ospedali/edifici sanitari diversi (es., ASL) utilizzando il continual learning e aggiornato nel tempo con l’online learning. I dati dal 2006 al 2018 sono stati utilizzati per la fase di training, mentre quelli del 2019 e 2020 sono stati utilizzati per la fase di testing.

Il modello è stato implementato in ambiente Python ed è compatibile con versioni 3.10. Le librerie utilizzate sono:

* numpy: pacchetto fondamentale per il calcolo scientifico in Python.
* pandas: libreria per la manipolazione e l'analisi dei dati.
* matplotlib: libreria completa per la creazione di visualizzazioni statiche, animate e interattive in Python.
* lightgbm: framework di gradient boosting progettato per l'addestramento distribuito ed efficiente di modelli di apprendimento automatico su larga scala.
* skforecast: libreria Python progettata per la previsione delle serie temporali.

Il flusso di lavoro è rappresentato nelle Figure successive. In particolare si prevede un’applicazione Software-as-a-Servive cloud su Microsoft Azure.



**RISULTATI**

L’algoritmo è stato allenato sul dataset a partire dal 1/1/2006 al 31/12/2018. L’auto-regressione è stata poi usata per predire il carico giornaliero nell’anno 2019. Una Bayesian search è stata implementata per selezionare gli iper-parametri ottimi per il modello.

I risultati sono mostrati in Figure 5. In particolare, sia il carico che la sua predizione sono stati riportati per apprezzare l’accuratezza del modello. Il massimo errore medio assoluto registrato è infatti inferiore al 5%.

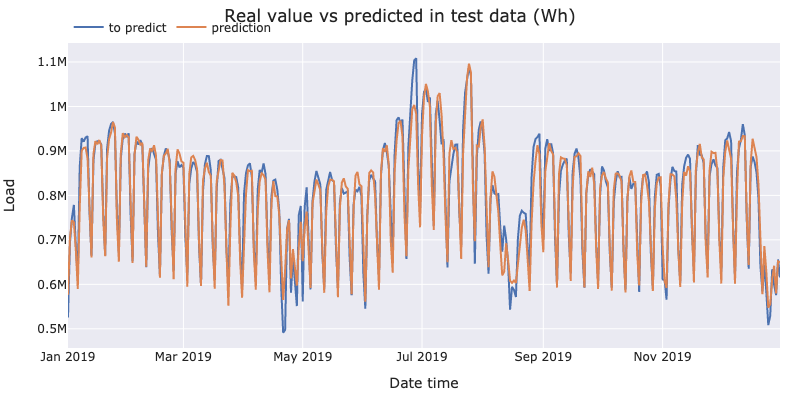


Figure 5. Predizioni prodotte dall’auto-regressione per l’anno 2019.

Questo risultato preliminare può essere migliorato incorporando nel modello variabili esogene (es., la temperatura e/o umidità dell’aria), la cui considerazione nella fase di training può notevolmente migliorare l’accuratezza predittiva del modello. Continual learning e online learning saranno poi implementati per favorire la scalabilità a numerose strutture ospedaliere/sanitarie. Una web application/dashboard sarà quindi realizzata per favorire la usabilità dell’algoritmo e provvedere a fornire analisi statistiche e di previsione in maniera efficiente.