Predicting Order Assignments Based on Historical Data

Azar Bagheri

***Abstract*—This project investigates a data-driven approach to automating the assignment of operators to incoming orders within a company that has systematically recorded two years of historical order data. Machine learning techniques were applied to develop predictive models aimed at identifying recurring patterns in assignment practices and accurately forecasting future operator assignments. This paper provides a structured overview of the data preprocessing methodologies, feature engineering processes, and machine learning models evaluated, including neural networks, decision trees, random forests, K-nearest neighbors, support vector machines, and recurrent neural networks. Results demonstrate moderate predictive performance across models, with certain algorithms exhibiting greater accuracy in capturing complex, non-linear assignment patterns within the data.**

I. Introduction

A. Background

The company currently manages operator assignment manually, with assignments rotating across a team depending on availability and work shifts. The current system is labor-intensive and lacks scalability. This project aims to develop a data-driven model to automate operator assignment, improving efficiency while reducing human involvement.

B. Objective

The objective is to analyze historical order data to predict the operator who will handle new orders based on patterns discovered in assignment trends. Various machine learning models, including both classical algorithms and deep learning techniques, were tested to identify the most accurate predictive approach.

II. Dataset

A. Data Overview

The dataset includes two years of historical order data containing:

- A unique identifier for each order.
- The timestamp of order completion.
- The operator ID who completed the order.

B. Data Quality and Challenges

The dataset required cleaning to handle missing values and standardize timestamp formats. Given the time-based nature of operator assignments, a significant challenge was identifying the rotational pattern in operator assignments across shifts and days.

III. Data Processing and Pipelines

A. Data Cleaning

The following steps were taken to preprocess the data:

- **Missing Values:** Rows with missing data were removed to ensure consistency.
- **Time Conversion:** Timestamps were converted into datetime format, enabling extraction of features like hour, day, and month.
- **Categorical Encoding:** Label encoding was applied to convert the target variable, operator_id, from categorical to numerical format.

B. Feature Engineering

Several temporal and categorical features were derived:

- **Time-based Features:** Hour, day, weekday, and month were extracted from timestamps.
- **Order Load Features:** The count of orders per operator per period was considered to capture operator efficiency.

The dataset was then split into training and testing sets (80% training, 20% testing) for model evaluation.

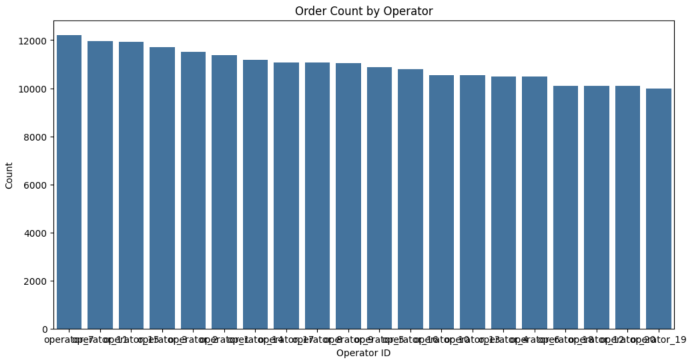

Fig. 1. operator assignment patterns

IV. Modeling Approach

Three primary Python scripts were used, each exploring a variety of machine learning models. Below is a breakdown of the different models and techniques used.

A. Script 1: Classical Models and Neural Network

Model Descriptions

1) **Neural Network:** A simple fully-connected neural network with two hidden layers (64 and 32 neurons) using ReLU activation, and a softmax output layer.

2) **Decision Tree Classifier:** A non-parametric model to capture decision boundaries based on feature splits.
3) **Random Forest Classifier:** An ensemble model with 100 estimators for improved generalization.
4) **K-Nearest Neighbors (KNN):** A distance-based model using the 5-nearest neighbors.

The Random Forest model achieved the highest accuracy, marginally surpassing the Decision Tree and KNN models, indicating moderate effectiveness in recognizing patterns in the dataset.

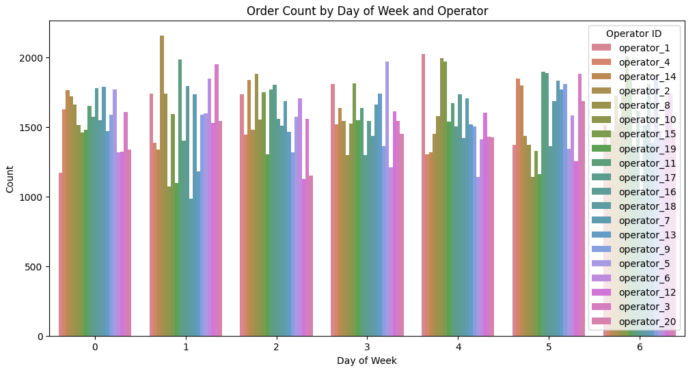

Fig. 2. Explore the data

B. Script 2: SVM and Expanded Neural Network

Model Descriptions

1) **Support Vector Machine (SVM):** A model for classification with a radial basis function (RBF) kernel to handle non-linear patterns which performed on par with other models.
2) **Neural Network:** A deep neural network with 32 and 16 neurons in hidden layers, trained for 50 epochs, achieving satisfactory performance though specific metrics were not specified.

C. Script 3: LSTM Model with Time-Series Data

Model Description

LSTM (Long Short-Term Memory): An RNN-based model to capture sequential order assignment patterns. It used two LSTM layers with leaky ReLU activation, specifically chosen for time-sequence data.

The LSTM model achieved a moderate accuracy score, indicating its utility in handling sequential data, though it was constrained by the dataset's complexity and required substantial training time to converge effectively.

V. Results and Discussion

A. Model Comparison

- The Random Forest model outperformed other classical models, achieving the highest accuracy among them, suggesting that its ensemble learning approach more effectively captured underlying patterns in the data.

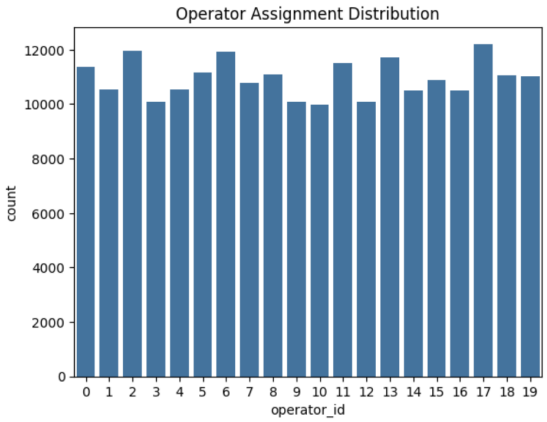

Fig. 3. order

- Neural Network accuracy varied significantly across scripts, with the model in Script 2 performing better after increased epochs.

The LSTM model proved effective for handling sequence-based data; however, it achieved only moderate accuracy, likely due to the limited sequential depth of the dataset. Additionally, the model required significantly more training time compared to other approaches.

B. Key Observations

Patterns in the dataset, such as variations in operator assignment by day of the week and workload, were identified and visualized. These visualizations highlighted areas where operator assignments varied significantly, suggesting potential temporal and workload-based influences.

C. Challenges and Limitations

A major limitation was the dataset's sequential depth and complex patterns in operator shifts. Further data or additional context, such as shift schedules, could enhance prediction accuracy.

VI. Conclusion

This project demonstrated the feasibility of predicting operator assignments using historical data and machine learning models. The Random Forest model showed the most promise among classical models, while the LSTM was effective in capturing sequential patterns, albeit with moderate accuracy. Further improvement could be achieved by incorporating additional features, such as operator shift data, and exploring more advanced sequential models.