

# Predicting Order Assignments Based on Historical Data

Azar Bagheri

**Overview:** This project aims to automate the assignment of operators to incoming orders using historical data and machine learning models. The current manual assignment system is labor-intensive, with assignments rotating among operators based on availability and work shifts. Using two years of order completion data, this project develops predictive models to identify patterns in operator assignments and forecast future assignments, ultimately improving operational efficiency.

**Dependencies and Installation:** To run the prediction script, ensure that Python (version 3.7 or higher) and the following libraries are installed: `pandas`, `numpy`, `scikit-learn`, `tensorflow/keras`, `matplotlib`, `seaborn`. Install dependencies using:

```
pip install pandas numpy scikit-learn tensorflow matplotlib seaborn
```

## Running the Script:

1. **Data Preparation:** Ensure the dataset file `order-dataset.csv` is in the same directory as the script or update the file path in the code.
2. **Run the Script:** Execute in a Python environment:  

```
python order_assignment_prediction.py
```
3. **Results:** The script outputs accuracy scores for each model and visualizations of operator assignment patterns.

**Approach:** The project employs a range of machine learning models to predict operator assignments:

- **Classical Models:** Random Forest, Decision Tree, K-Nearest Neighbors, Support Vector Machines for static and non-linear patterns.
- **Neural Networks:** Fully connected deep networks for general pattern recognition.
- **LSTM:** A recurrent neural network model designed for time-sequence data to capture sequential order assignment patterns.

Models are trained on preprocessed features from timestamps (hour, day, month), and performance is evaluated using accuracy metrics. The Random Forest model showed the highest accuracy, while the LSTM captured sequential dependencies but required significant training time.