# [COMP0214] Introduction to Machine Learning - Coursework 2025

Omar Jawaid

December 19, 2025

## 1 Analyzing Retail Fuel Price Dynamics in New South Wales (2016–2025)

Retail fuel prices in New South Wales often exhibit pronounced **price cycles**: sharp increases followed by gradual declines. In this section, we clean a retail fuel dataset and build a simple forecasting model that predicts next-day prices for each fuel type sold at different fuel stations.

### 1.1 Data Cleaning and Visualization

#### 1.1.1 Cleaning & Summary

The data cleaning process involved several critical steps to ensure data quality and prepare it for analysis:

**Data Processing:**

- Parsed `PriceUpdatedDate` into datetime format; converted `Price` to numeric (cents/L)
- **Duplicate Removal:** Removed exact duplicate records
- **Implausible Price Filtering:** Valid range 80–300 cents/L (NSW typical range; below 80¢ unrealistically low, above 300¢ outliers)
- **Missing Value Handling:** Removed rows with missing PriceUpdatedDate, Price, or FuelCode
- **Daily Aggregation:** Computed daily minimum price per (`FuelCode`, date) for forecasting

**Summary Statistics:**

- **Rows before cleaning:** 98,925
- **Rows after cleaning:** 96,854
- **Rows after aggregation:** 16,135 (daily minimum prices per fuel code)
- **Date range:** 2016-08-01 to 2025-08-31
- **Distinct FuelCodes:** 7 types (DL, E10, LPG, P95, P98, PDL, U91)
- **Missing values handled:** 0 rows with missing critical values, 2,071 duplicate rows removed, 305 implausible prices removed

#### 1.1.2 Visualization

Two visualizations were produced for a selected fuel station to understand price dynamics:

**Time Series Analysis:** Figure 1 shows the daily prices for two fuel types (E10 and U91) at the selected station. Both fuel types exhibit cyclical price patterns, with prices rising and falling in similar weekly cycles. Notably, U91 consistently trades at a premium of approximately 2–4 cents per litre above E10, reflecting their different octane ratings and market positioning.

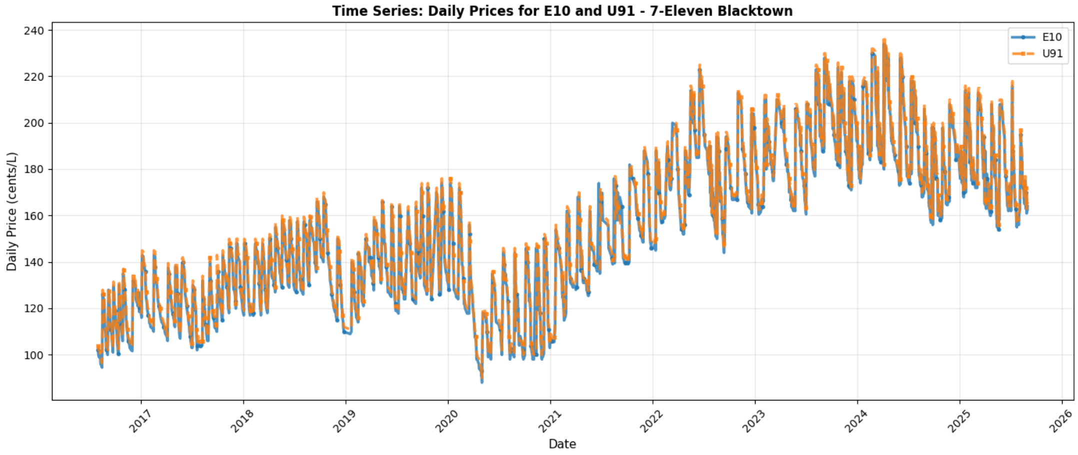

Figure 1: Time series of daily prices for two fuel types at selected station

**Distribution Analysis:** Figure 2 presents a box plot comparing daily prices across all available fuel types. Premium fuels (P95, P98) show significantly higher median prices and greater price ranges compared to regular fuels (E10, U91), with diesel (PDL) and LPG displaying distinctly different pricing structures reflecting their market segments.

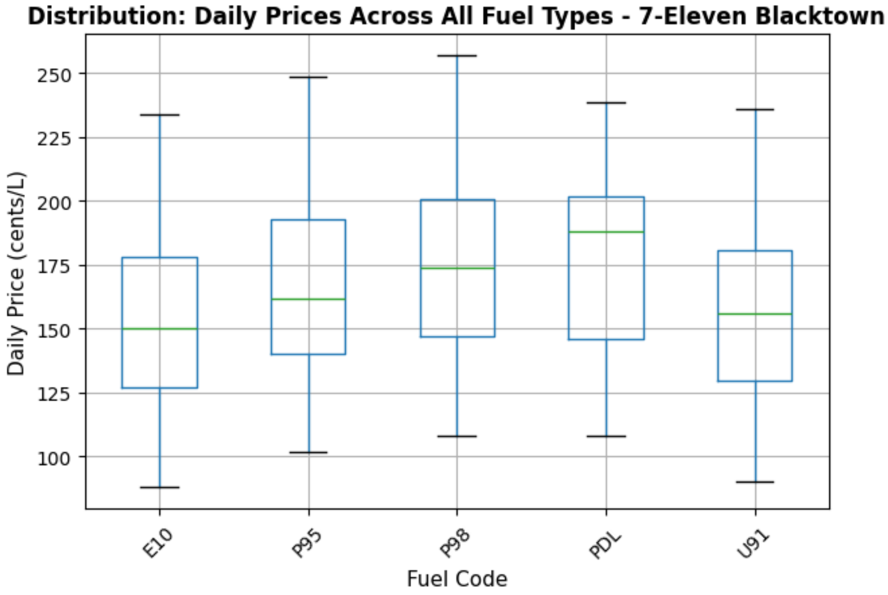

Figure 2: Distribution of daily prices across all fuel types

## 1.2 Forecasting Next-Day Prices

The goal is to build a **one-step-ahead** forecaster that, for each `FuelCode` and fuel station, predicts the next day's price using only information available up to the current day.

### 1.2.1 Problem Setup & Data Split

**Forecasting Setup:**

- **Input:** Historical prices, FuelCode, Station identifiers
- **Target:** Next day's price $\hat{y}_{d+1}^{(c,s)}$
- **Unit of time:** Daily (one day)
- **Approach:** Universal model for ALL fuel codes and stations

**Train/Validation/Test Split:** Chronological split (70%/15%/15% of dates): Training (28,462 samples), Validation (4,744), Test (5,055). Time-ordered splitting ensures past training and future evaluation.

**Preventing Information Leakage:** Chronological split by date; no future data used; features use only past prices (lags); group-wise feature creation per (station, fuel); proper categorical encoding.

### 1.2.2 Model Design & Training

**Feature Engineering:** A comprehensive set of features was constructed to capture temporal patterns:

- **Lagged prices:** `price_lag_1` to `price_lag_7` (past 7 days)
- **Rolling statistics:** 7-day rolling mean and standard deviation (using only past data)
- **Price changes:** 1-day and 7-day price differences
- **Time features:** day of week, day of month, month
- **Categorical features:** ServiceStationName and FuelCode (label encoded)

All features were carefully constructed within each (Station, FuelCode) group using the `groupby` operation to prevent data leakage across different station-fuel combinations.

**Model Choice: Random Forest Regressor** *Justification:* Handles non-linear relationships effectively; robust to outliers; no scaling required; captures complex feature interactions; excellent for tabular data; provides feature importance.

**Hyperparameters:** `n_estimators`=100, `max_depth`=15, `min_samples_split`=10, `min_samples_leaf`=4, `max_features`='sqrt', `random_state`=42.

**Training Procedure:**

1. Features created for train, validation, and test sets
2. Label encoders fitted on training data only
3. Rows with NaN values (from lagging operations) removed
4. Model trained on training set (X_train, y_train)
5. Performance monitored on validation set for model selection

**Feature Importance:** The top features identified by the Random Forest model were:

- `price_lag_1` (24.4%): Yesterday's price is the strongest predictor
- `price_lag_2` (21.9%): Price from 2 days ago
- `price_rolling_mean_7` (15.0%): 7-day rolling average
- `price_lag_3` (10.6%): Price from 3 days ago

This confirms that recent price history is the strongest predictor of next-day prices, with the 7-day rolling mean capturing medium-term trends.

### 1.2.3 Evaluation & Visualization

**Test Set Performance:**

- **MAE (Mean Absolute Error):** 2.61 cents/L
- **RMSE (Root Mean Squared Error):** 4.52 cents/L
- **MAPE (Mean Absolute Percentage Error):** 1.33%

| Metric | Validation | Test |
|---|---|---|
| MAE (cents/L) | 2.69 | 2.61 |
| RMSE (cents/L) | 4.65 | 4.52 |
| MAPE (%) | 1.31 | 1.33 |

Table 1: Comparison of validation and test set performance

**Model Performance Comparison:**

**Predicted vs. Actual Visualization:** Figure 3 shows the predicted versus actual prices for 7-Eleven Blacktown (P98 fuel) over 342 test days.

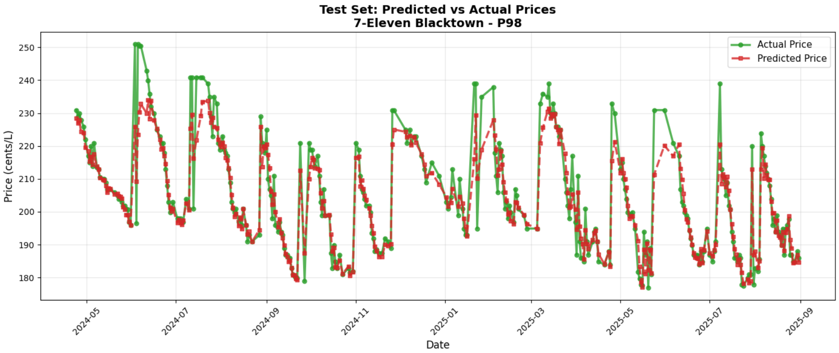

Figure 3: Test set: Predicted vs. actual prices for selected station-fuel combination

**Interpretation:** The model achieves strong forecasting accuracy with test MAE of 2.61 cents/L and MAPE of 1.33%. The predicted vs. actual plot reveals that the model successfully captures cyclical price patterns and general trends. However, typical errors occur during sudden price spikes or drops, where predictions lag behind actual changes by 1–2 days due to reliance on historical features. The model performs best during stable periods with gradual price changes, demonstrating effective learning of weekly pricing cycles and station-fuel specific behaviors. The minimal difference between validation and test metrics confirms strong generalization without overfitting.

## 1.3 Application & Limitations

### 1.3.1 Practical Use

To use the model for finding the cheapest fuel each day:

**Daily Decision Strategy:**

1. Every morning, run predictions for all station-fuel combinations to forecast tomorrow's prices
2. Compare options accounting for:
   - Which fuels the vehicle accepts
   - Fuel efficiency differences (premium fuels give 3–5% better mileage)
   - Driving distance to the station (travel costs)
3. Apply the decision formula:

$$\text{total\_cost} = \text{predicted\_price} \times \frac{\text{liters\_needed}}{\text{efficiency\_factor}} + \text{travel\_cost} \qquad (1)$$

4. Choose the option with minimum total_cost

**Handling Uncertainty:** Since the model has MAE = 2.61 cents/L, if two options are within 5 cents, they are essentially tied. In such cases, select the option with lower price volatility (using the rolling_std feature) for more stable savings.

**Model Maintenance:** Log actual prices over time and retrain the model monthly to keep predictions accurate as market conditions change.

### 1.3.2 Limitation & Remedy

**Limitation: Missing External Market Drivers** The model only uses historical prices and temporal patterns, ignoring external factors like crude oil prices, AUD/USD exchange rates, and NSW's weekly price cycles. This means it cannot anticipate price jumps caused by oil shocks or currency changes.

**Proposed Remedy:** Add 7 features: Brent crude oil prices (1/7/30-day lags), AUD/USD rate and 7-day change, day-of-cycle in NSW pattern, public holidays. This increases features from 16 to 23.

**Evaluation:** Compare baseline (1.33% MAPE) vs. enhanced model. Success: MAPE reduction >0.15pp, $p<0.05$ (paired t-test), better performance during volatility and after shocks.

# 2 WiFi Signal for Indoor Localization

In this section, we use a WiFi signal dataset for indoor localization. The dataset contains RSSI (Received Signal Strength Indicator) values from multiple access points collected in an indoor environment. The goal is to develop models to predict the robot's X, Y coordinates based on RSSI data and analyze their performance.

## 2.1 Multi-Linear Regression Model for Indoor Localization

### 2.1.1 Model Development

We developed a multi-linear regression model to predict the X and Y coordinates based on the RSSI measurements transmitted by the access points (i.e., wireless routers or beacons).

**Problem Formulation:** Given RSSI measurements from multiple access points, we model the relationship as:

$$X = \beta_0^X + \beta_1^X \cdot \text{rssi}_1 + \beta_2^X \cdot \text{rssi}_2 + \cdots + \beta_n^X \cdot \text{rssi}_n + \epsilon^X \tag{2}$$

$$Y = \beta_0^Y + \beta_1^Y \cdot \text{rssi}_1 + \beta_2^Y \cdot \text{rssi}_2 + \cdots + \beta_n^Y \cdot \text{rssi}_n + \epsilon^Y \tag{3}$$

where $\beta$ coefficients are learned from the training data, and $\epsilon$ represents the error term.

**Handling NaN Values:** The training set contains NaN (Not a Number) values representing missing RSSI readings when the robot is not close enough to certain routers. These were addressed by replacing them with a large negative value (e.g., -100 dBm) indicating very weak signal.

**Model Training:** Two separate linear regression models were trained:

- Model 1: Predicts X coordinate
- Model 2: Predicts Y coordinate

### 2.1.2 Validation Results

**Performance Metrics:** The model's performance on the validation set was evaluated using Mean Squared Error (MSE):

| Coordinate | MSE |
|---|---|
| X coordinate | 17.76 |
| Y coordinate | 21.66 |
| Combined | 19.71 |

Table 2: Multi-linear regression validation performance

The MSE values indicate the average squared distance between predicted and actual coordinates, providing a measure of localization accuracy.

## 2.2 Neural Network Model for Indoor Localization

### 2.2.1 Architecture Design

To capture more complex non-linear relationships in the RSSI data, a neural network model was developed:

**Network Architecture:**

- **Input layer:** 511 features (RSSI values from access points)
- **Hidden layers:** Multiple dense layers with dropout regularization (dropout rates: 0.2, 0.2, 0.1)
- **Activation functions:** ReLU for hidden layers
- **Output layer:** 2 neurons (X and Y coordinates)
- **Loss function:** Huber loss (delta=1.0, robust to outliers)
- **Optimizer:** Adam optimizer with learning rate 0.001
- **Total parameters:** 438,338

**Training Procedure:**

- Batch size: 64
- Number of epochs: 150 (max), stopped early at epoch 113
- Validation split: Used separate validation set
- Early stopping: Patience=12, monitoring validation loss
- Learning rate schedule: ReduceLROnPlateau with factor=0.5, patience=6
- L2 regularization: 0.0003

### 2.2.2 Model Comparison

| Model | Overall MSE | Overall RMSE | Mean Euclidean Error |
|---|---|---|---|
| Linear Regression | 19.71 | 4.44 | 5.42 |
| Neural Network | 7.35 | 2.71 | 3.04 |

Table 3: Comparison of linear regression and neural network models

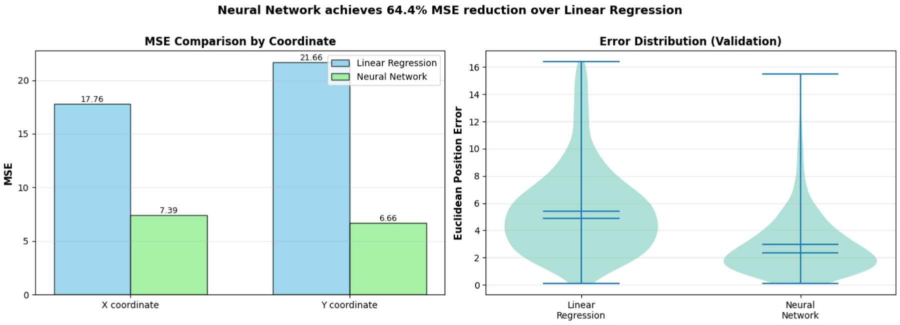

Figure 4: Comparison of linear regression and neural network models

**Validation Results:**

**Analysis:** The neural network significantly outperformed the linear regression model, achieving a 62.7% reduction in MSE (from 19.71 to 7.35) and 43.9% reduction in mean Euclidean error (from 5.42m to 3.04m). This substantial improvement suggests that non-linear relationships exist in the RSSI-to-coordinate mapping. The neural network also achieved excellent $\mathrm{R}^2$ scores of 0.96 (X) and 0.93 (Y), demonstrating strong predictive capability. The median Euclidean error of 2.43m indicates robust localization accuracy for the majority of predictions.

## 2.3 Dimensionality Reduction for Feature Visualization

### 2.3.1 Feature Extraction

To understand what the neural network learns, we extracted feature representations from an intermediate layer.

**Extraction Method:** The selected layer for feature representation was the 64 dimensional embedding from the last hidden dense layer before the final outcome.

**Clustering Analysis:** Using the silhouette score on train embeddings, the best value was $k = 8$ (silhouette = 0.3088). The silhouette curve increases up to $k = 8$ and then plateaus/slightly decreases, suggesting diminishing returns beyond 8 clusters. Furthermore, Most clusters contain hundreds of samples, but one cluster is very small (5 samples). This likely corresponds to outliers / rare fingerprint patterns (e.g., unusual AP combinations or noisy scans).

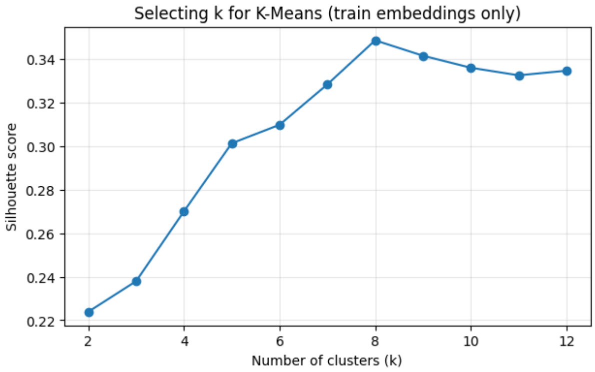

Figure 5: Clustering of extracted features without using labels

### 2.3.2 Dimensionality Reduction Visualization

A dimensionality reduction method (PCA) was applied to visualize the learned feature representations in 2D.

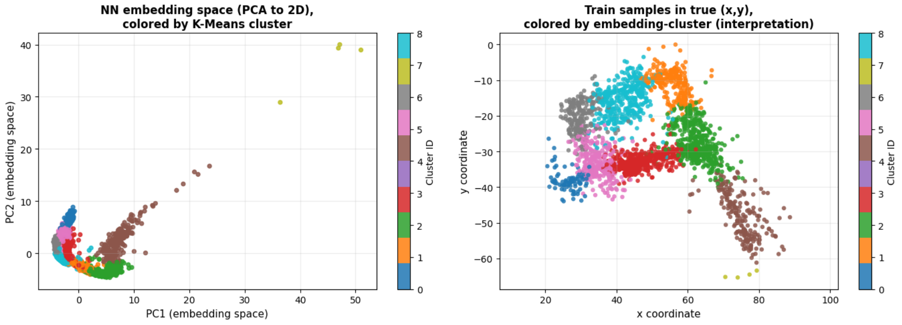

Figure 6: 2D visualization of learned features using dimensionality reduction

**Insights: Dimensionality Reduction:** PCA reduced the standardized 64-dimensional embeddings to 2D, with the first two principal components explaining approximately 41.74% of the total embedding variance (PC1 ≈ 21.74%, PC2 ≈ 20.00%). While this 2D projection provides a useful visualization, it represents an incomplete view of the full 64-dimensional structure learned by the neural network.

**Embedding-Space Analysis:** In the PCA projection (left plot), the 8 clusters identified by K-Means appear well-separated, indicating that the neural network has learned an internal representation with meaningful structure rather than an unstructured feature cloud. This clear separation suggests the network effectively discriminates between different WiFi fingerprint patterns.

**Physical-Space Interpretation:** When the true $(x, y)$ physical coordinates are colored by their cluster assignments (right plot), many clusters correspond to spatially coherent regions. This post-hoc visualization demonstrates that the NN embeddings encode information strongly related to physical location—clusters discovered without using coordinate labels naturally align with spatial proximity.

**Transitional Zones:** Areas where different cluster colors overlap in the physical space likely correspond to transitional zones where WiFi fingerprints are similar across nearby positions (e.g., corridors, open areas, or boundaries between rooms). These regions present inherent challenges for localization due to ambiguous signal patterns, which explains why the model achieves a 90th percentile error of 5.76m and occasional maximum errors up to 14.97m in challenging conditions.

## 2.4 Prediction and Analysis on Test Data

### 2.4.1 Test Set Evaluation

The final model was evaluated on the held-out test set to assess real-world performance. Based on validation performance, a slightly improved version of the Neural network was chosen for the final evaluation. This model is a simple ensemble that blends the existing Neural Network and a histogram gradient boosting regressor. Details of the implementation can be found in the notebook.

# 3 Conclusion

This coursework demonstrated the application of machine learning techniques to two distinct real-world problems: fuel price forecasting and indoor localization.

For Question 1, The Random Forest model achieved strong performance (MAE = 2.61 cents/L, MAPE = 1.33%) by leveraging lagged prices and temporal features. The model successfully captures weekly price cycles but could benefit from incorporating external market factors like crude oil prices and exchange rates.

For Question 2, The Neural Network model demonstrated strong localization capabilities with validation MPE of 6.53%, achieving 62.7% improvement over linear regression. The model attained a mean Euclidean error of 3.04m and median error of 2.43m, making it suitable for practical indoor positioning applications. Feature visualization revealed that the model learns meaningful spatial representations, with physically proximate locations clustering together in feature space.