WiDS Kalman Filtered Trend Trader

Assignment 1

Lavanya Padole 24B1292

February 2, 2026

1 Question 1: Linear Regression

1.1 Model Definition

The multiple linear regression model with p predictors is written as:

$$y = X\beta + \epsilon,$$

where $X \in \mathbb{R}^{n\times(p+1)}$ is the design matrix with an intercept column, $\beta = (\beta_0, \beta_1, \ldots, \beta_p)^\top$ is the parameter vector, and $\epsilon \sim \mathcal{N}(0, \sigma^2 I)$ is the noise term.

1.2 OLS Objective and Solution

Ordinary Least Squares minimizes the Mean Squared Error:

$$J(\beta) = \frac{1}{n}\|X\beta - y\|^2$$

Taking the gradient and setting it to zero yields:

$$\hat{\beta} = (X^\top X)^{-1} X^\top y$$

1.3 Numerical Results

The estimated OLS coefficients using NumPy are:

$$\hat{\beta} = \begin{bmatrix} -0.075997 \\ 3.019213 \\ -1.998648 \\ 0.465321 \\ 0.028746 \\ 0.978880 \\ 0.005197 \\ 0.050117 \\ -0.917746 \\ 1.951978 \\ -0.012285 \\ 0.057589 \\ 0.027415 \end{bmatrix}$$

These coefficients exactly match those obtained using `sklearn.linear_model.LinearRegression`, confirming the correctness of the closed-form OLS implementation.

1.4 Residual Diagnostics

Residuals versus fitted values and a Q–Q plot were analyzed. The residuals show no strong systematic pattern, suggesting approximate homoscedasticity. The Q–Q plot indicates mild deviations from normality in the tails, which is common in practical regression problems but does not severely violate model assumptions.

1.5 Leverage and Influence

The hat matrix was computed as:

$$H = X(X^\top X)^{-1}X^\top$$

The top ten high-leverage observations were:

$$\{212, 168, 241, 124, 39, 293, 112, 240, 160, 96\}$$

Cook's distance identified the following influential points:

$$\{155, 296, 267, 59, 286, 117, 86, 146, 293, 116\}$$

These observations have the greatest potential impact on regression coefficients. Not all high-leverage points are influential, highlighting the importance of considering both leverage and residual magnitude.

2 Question 2: Salary Prediction and Bias Detection

2.1 Exploratory Data Analysis

The dataset contains 12,000 employee records. The mean salary is \$105,862 with a standard deviation of \$28,251, indicating substantial salary dispersion and the presence of high-income outliers. Salary distributions differ across gender categories as observed from boxplots.

2.2 Preprocessing

Missing values were removed to ensure numerical stability. Categorical features were encoded using one-hot encoding with a reference category dropped to avoid multicollinearity. A stratified train-test split was performed using gender to preserve group proportions.

2.3 Model Training and Evaluation

An OLS Linear Regression model was trained.

- RMSE: **17,582.01**
- MAE: **7,465.70**
- R^2: **0.629**

The model explains approximately 63% of the variance in employee salaries.

2.4 Fairness Analysis

Gender was treated as the protected attribute.

Metric	Male vs Female	Male vs Other
Mean Prediction Difference	5,225.21	1,405.19
MAE Difference	-179.18	966.34

Male employees receive higher predicted salaries on average. However, the MAE difference between male and female employees is small, suggesting similar prediction accuracy across groups.

2.5 Residual Bias and Statistical Testing

A two-sample t-test on residuals between male and female groups produced a p-value of 0.064. At the 5% significance level, this result indicates no statistically significant difference in mean residuals, suggesting the model does not systematically overestimate or underestimate salary for either group.

3 Question 3: Deep Neural Network Classifier

3.1 Model Architecture

A deep neural network named `DigitClassifier` was designed with the following structure:

- Input layer of dimension 784
- Hidden layer 1: 256 neurons with ReLU activation
- Hidden layer 2: 128 neurons with ReLU activation
- Output layer: 10 logits corresponding to digit classes

3.2 Training Setup

The model is trained for five epochs using the Adam optimizer with a learning rate of 0.001 and CrossEntropyLoss. A batch size of 64 is used. The training loop consists of forward pass, loss computation, backward propagation, and optimizer updates.

3.3 Why ReLU is Preferred

ReLU is preferred over Sigmoid and Tanh for deep networks for two primary reasons:

- ReLU mitigates the vanishing gradient problem by maintaining non-saturating gradients for positive inputs.
- It is computationally efficient, enabling faster convergence during training.

3.4 Role of Autograd

PyTorch's autograd engine automatically tracks tensor operations to construct a dynamic computational graph during the forward pass. During backpropagation, autograd applies the chain rule to compute gradients of the loss with respect to model parameters, enabling efficient and flexible gradient-based optimization.

3.5 Implementation Note

The PyTorch implementation is provided as requested. Model execution was not performed, as the assignment assumes the availability of predefined data loaders.