Date: 02-11-2025 **Name:** Harini LV
Assignment: 1 **Roll No:** 3122237001016

CAT-1 THEORY ASSIGNMENT

Course: Machine Learning Algorithms

GitHub Repository:

https://github.com/HariniLV-3103/Machine-Learning/tree/main/Theory%20Assignment

---

1. Regression — Mobile Phone Price Prediction

Aim

To create and assess matrix-based linear regression models for mobile phone pricing prediction, such as gradient descent, L2-regularized (ridge), and closed-form techniques and compare their performance with and without feature standardisation.

Objectives

1. Implement closed-form and gradient descent regression models.
2. To perform ridge regression and L2 regularisation.
3. Evaluate the model performance with and without standardized features.
4. Visualize the relationship between predicted and actual prices.
5. Use regression coefficients to analyse feature importance.

Methodology

Mathematical Model: Let the data matrix X $(n \times (d+1))$ include a bias term, and let y denote the target vector.

$$\hat{y} = X\theta, \quad \text{where} \quad \theta = (X^T X)^{-1} X^T y$$

For ridge regression, the parameter vector is estimated as:

$$\theta_{\text{ridge}} = (X^T X + \lambda I)^{-1} X^T y$$

For gradient descent, the parameter update rule is given by:

$$\theta := \theta - \alpha \frac{1}{n} X^T (X\theta - y)$$

Implementation Steps

1. Load and preprocess the dataset by handling missing values and selecting relevant features.
2. Split the data into training and testing sets (80–20 ratio).
3. Construct the design matrix and target vector.

4. Implement the closed-form regression model and evaluate its performance using suitable metrics.
5. Implement gradient descent and visualize its convergence behavior.
6. Add L2 regularization and experiment with different values of λ.
7. Compare the performance of models with and without feature standardization.
8. Plot predicted prices against actual prices to assess model accuracy.
9. Analyze the significance of each feature using standardized regression coefficients.

Code

GitHub Link: https://github.com/HariniLV-3103/Machine-Learning/blob/main/Theory Assignment/Linear Regression.ipynb

Results and Analysis

Table 1: Performance Comparison of Linear Regression Methods

Model	MSE	RMSE	MAE	R^2
Closed-form OLS	23062.23	151.86	130.32	0.9593
Gradient Descent (no reg)	23167.57	152.21	126.35	0.9591
Ridge Closed-form ($\lambda = 1.0$)	23254.37	152.49	129.46	0.9590
Ridge Gradient Descent ($\lambda = 1.0$)	24160.31	155.43	124.89	0.9574
Ridge (standardized features)	23254.37	152.49	129.46	0.9590
Ridge (unstandardized features)	23107.47	152.01	130.04	0.9592

Interpretation: The closed-form model achieved strong performance ($R^2 \approx 0.96$). Unregularized GD diverged until features were standardized and the learning rate tuned. Ridge regression improved numerical stability and slightly reduced error at optimal λ values.

Predicted vs Actual Visualization

Scatter plots of predicted vs. actual prices showed tight clustering around the 45° line for OLS and tuned GD. Ridge regression slightly compressed predictions toward the mean for larger λ.

Effect of Standardization

Without standardization, large-magnitude features dominated the penalty term in ridge regression, leading to uneven regularization. Standardization yielded balanced coefficients and faster GD convergence.

Feature Importance

Coefficients from the standardized ridge model indicated the most influential features were RAM, processor speed, and battery capacity. L2 regularization shrunk less important feature weights toward zero.

Inference

- Linear regression successfully modeled price prediction with high accuracy.
- Gradient descent required feature scaling for stable convergence.
- Ridge regression enhanced robustness and generalization.
- Standardization was crucial for balanced regularization effects.

Learning Outcomes

- Understood matrix-based regression formulations.
- Differentiated between closed-form and iterative optimization.
- Learned importance of regularization and data scaling.
- Evaluated performance using MSE, RMSE, MAE, and R^2.
- Interpreted feature importance from standardized coefficients.

Conclusion

The matrix-based approach (OLS and Ridge) effectively predicted mobile phone prices with minimal error. Feature standardization and moderate regularization yielded optimal generalization. Gradient descent was computationally efficient for large data, whereas closed-form solutions provided exact results for moderate-sized datasets.

2. Linear Classification — Bank Note Authentication

Aim

To fit and evaluate a Linear Classification Model (single or multi-layer neural network) for the **Bank Note Authentication dataset**, and analyze the suitability of linear models for binary classification.

Objectives

1. Divide the dataset into training and testing sets.
2. Fit classification models with and without L2 regularization; compare accuracies.
3. Plot training and test accuracy versus λ.
4. Visualize classification in 3D using three important features.
5. Introduce artificial outliers and observe their effect.
6. Refit the classifier and analyze the impact on model performance.

Description

The Bank Note Authentication dataset contains statistical features derived from banknote images. The task is to classify notes as genuine or forged. Each instance is characterized by four continuous features: *variance, skewness, kurtosis, and entropy.*

Methodology

A logistic regression classifier was implemented using both unregularized and L2-regularized versions. The dataset was split into 80% training and 20% testing subsets. The training process was repeated for various λ (regularization strengths) to analyze bias–variance behavior.

$$J(w) = -\frac{1}{n}\sum_{i=1}^{n}\left[y_i \log(\hat{y}_i) + (1-y_i)\log(1-\hat{y}_i)\right] + \frac{\lambda}{2n}\|w\|^2$$

Implementation Steps

1. **Data Loading:** Imported the Bank Note Authentication dataset (CSV format) containing features — variance, skewness, kurtosis, and entropy — along with binary labels (0 = forged, 1 = genuine).
2. **Preprocessing:** Checked for missing values and standardized all features using `StandardScaler` to ensure uniform scale across features.
3. **Train-Test Split:** Split the dataset into 80% training and 20% testing sets using `train_test_split()` with a fixed random seed for reproducibility.
4. **Model Training (No Regularization):** Trained a logistic regression classifier without regularization to establish a baseline accuracy.

5. **Model Training (With L2 Regularization):** Re-trained the model with different regularization strengths ($\lambda = 0.01, 0.1, 1, 10$) and observed the changes in training and test accuracies.
6. **Accuracy vs. λ Plot:** Plotted both training and test accuracies against log-scaled λ values to analyze underfitting and overfitting trends.
7. **3D Visualization:** Selected three key features and visualized data points in 3D space using color-coded class labels to examine linear separability.
8. **Outlier Injection:** Introduced artificial outliers by adding a fixed offset to a subset of training samples and retrained the classifier to study robustness.
9. **Evaluation:** Compared accuracies, confusion matrices, and decision boundary plots between normal and outlier-injected data.
10. **Documentation:** All implementation steps, results, and plots were consolidated in the project repository (link placeholder below).

Code

GitHub Link: https://github.com/HariniLV-3103/Machine-Learning/blob/main/Theory Assignment/Linear Classification.ipynb

Results and Analysis

- **Train-Test Split:** 80–20 random split ensured balanced class distribution.
- **Without Regularization:** Accuracy $\approx 99.2\%$ (high fit, possible overfitting).
- **With L2 Regularization:** Optimal $\lambda = 0.1$ yielded $\approx 98.8\%$ accuracy.
- **Accuracy vs. λ Plot:** Accuracy remained stable for small λ but decreased for larger values due to underfitting.

3D Visualization

A 3D scatter plot of the first three features (variance, skewness, kurtosis) was plotted with class labels. The classes were linearly separable, confirming the suitability of linear models.

Outlier Analysis

Data points were intentionally shifted (adding a large offset to a subset of features). The classifier's performance dropped slightly ($\sim$3–5%) and decision boundary became less stable. Regularization reduced sensitivity to these outliers.

Inference

Linear classification proved effective for this dataset due to near-linear separability. L2 regularization slightly improved robustness. However, the model's sensitivity to outliers demonstrates the need for preprocessing and possibly more robust algorithms for noisier data.

Learning Outcomes

- Understood the role of regularization in controlling overfitting.
- Visualized data separability and impact of outliers.
- Evaluated performance of linear classifiers under varying regularization.

Conclusion

The Linear Classification experiment on the Bank Note Authentication dataset demonstrated that logistic regression is highly effective for binary classification when data exhibits near-linear separability. Regularization played a key role in preventing overfitting without significantly impacting accuracy. The introduction of outliers slightly degraded performance, confirming that while linear classifiers are computationally efficient, their decision boundaries are sensitive to data perturbations. Overall, the experiment reinforced the theoretical and practical understanding of:

- How L2 regularization stabilizes learning.
- The importance of scaling and clean data for linear models.
- The interpretability and efficiency of logistic regression for binary problems.