Assignment 2
Training Data Influence Analysis

Introduction

This report presents an analysis of the influence of training data on the performance of a machine learning model. The goal is to understand which groups of observations or individuals are "responsible" for a given model output or capability. The analysis is performed on the **Adult Income Dataset**, a binary classification task predicting whether an individual earns more than $50K/year.

---

1. Preliminaries

1.1 Dataset Description

The **Adult Income Dataset** is sourced from the UCI Machine Learning Repository. It contains information about individuals, including demographic and employment-related features, and the target variable is binary:

- **Target Variable**: income (0 for <=50K, 1 for >50K).
- **Features**: Age, workclass, education, marital status, occupation, relationship, race, sex, capital gain, capital loss, hours per week, and native country.
- **Dataset Size**: 48,842 rows and 14 columns.

1.2 Preprocessing

The dataset was preprocessed as follows:

1. **Handling Missing Values**: Rows with missing values were dropped.
2. **Encoding Categorical Variables**: Categorical features (e.g., workclass, education) were one-hot encoded.
3. **Normalizing Numerical Features**: Numerical features (e.g., age, hours per week) were normalized using StandardScaler.
4. **Train-Test Split**: The data was split into training (80%) and test (20%) sets using a random seed for reproducibility.

1.3 Baseline Model

A **logistic regression model** was chosen as the baseline classifier due to its simplicity and interpretability. The model was trained with the following hyperparameters:

- max_iter=1000: To ensure convergence.
- class_weight='balanced': To handle class imbalance.

1.4 Model Performance

The baseline model achieved the following performance on the test set:

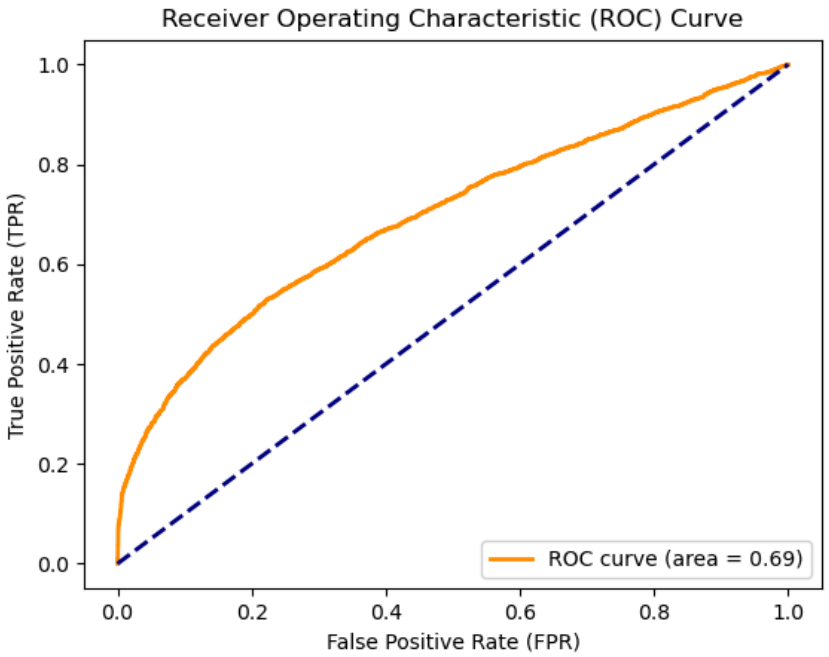
- **Accuracy**: 0.6416
- **ROC AUC**: 0.6947
- **Confusion Matrix:**

3304	**1632**
1869	**2964**

- **Classification Report:**

	precision	recall	f1-score	support
0	**0.64**	**0.67**	**0.65**	**4936**
1	**0.64**	**0.61**	**0.63**	**4833**
accuracy			**0.64**	**9769**
macro avg	**0.64**	**0.64**	**0.64**	**9769**
weighted avg	**0.64**	**0.64**	**0.64**	**9769**

The **ROC curve** for the baseline model is shown below:



The **Area Under the ROC Curve (AUC)** is **0.69**, indicating that the model has moderate discriminatory power. The curve shows the trade-off between the True Positive Rate (TPR) and False Positive Rate (FPR) at different classification thresholds.

The **primary metric** chosen for this analysis is **accuracy**, as it provides a straightforward measure of overall model performance.

---

2. Brute Force LOO Influence

2.1 Methodology

The Leave-One-Out (LOO) influence of 10 randomly selected training points was computed. For each point:

1. The point was removed from the training set.
2. The model was retrained on the reduced dataset.
3. The change in accuracy on the test set was recorded as the influence score.
4.

2.2 Results

The LOO influence scores for the 10 selected points are shown below:

Point Index	Influence Score
37272	-0.0003
1912	-0.0003
27220	-0.0003
33245	-0.0002
27732	-0.0003
27795	-0.0003
5715	-0.0001
2949	-0.0002
3818	-0.0003
35374	-0.0003

2.3 Analysis

All influence scores are small and negative, indicating that removing any of these points slightly **decreases** the model's accuracy. This suggests that these points are **not highly influential** in improving the model's performance. Instead, they contribute positively to the model's learning, as their removal harms performance.

---

3. Group-Level Influence

3.1 Methodology

The influence of 10 groups of training points was computed. Each group was randomly selected, with sizes ranging from 10% to 50% of the training data. For each group:

1. The group was removed from the training set.
2. The model was retrained on the reduced dataset.
3. The change in accuracy on the test set was recorded as the group influence score.

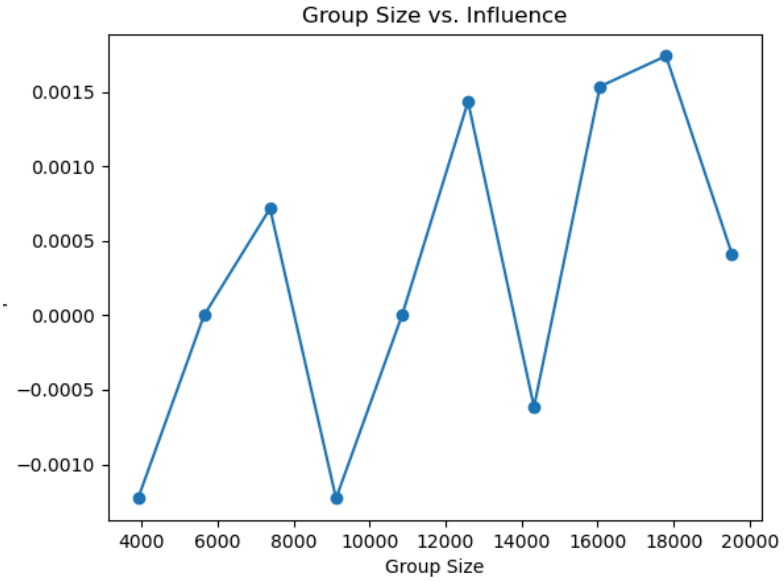
3.2 Results

The group influence scores are shown below:

Group Size	**Influence Score**
3907	-0.0012
5643	0.0000
7380	0.0007
9117	-0.0012
10853	0.0000
12590	0.0014
14326	-0.0006
16063	0.0015
17799	0.0017
19536	0.0004

4.3 Analysis

The plot below shows the relationship between group size and influence:



- Larger groups tend to have slightly higher influence scores, indicating that removing more data has a more noticeable impact on model performance.
- Some groups have positive influence scores, suggesting that they may contain noisy or redundant data.

---

4. Shapley Values

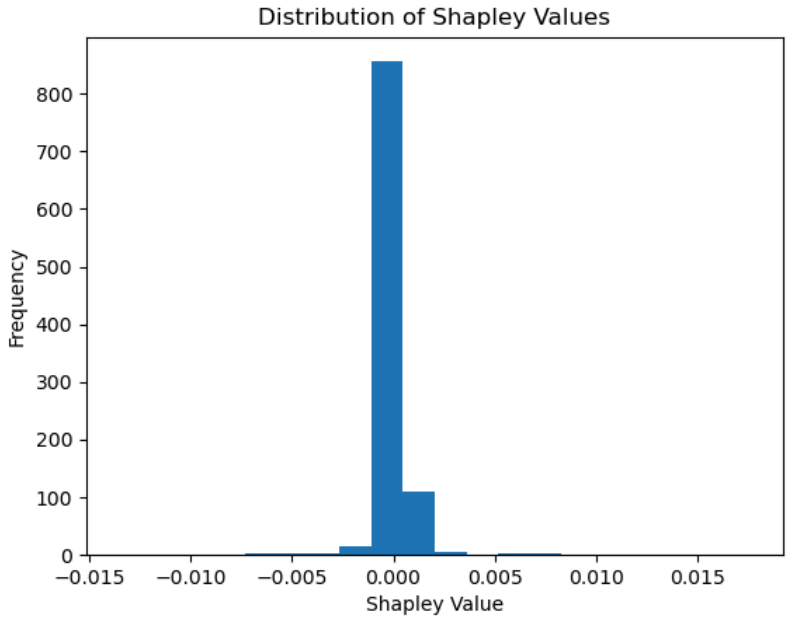
4.1 Methodology

Shapley values were estimated using **Truncated Monte Carlo Shapley Value Estimation**. The algorithm involves:

1. Randomly permuting the training data.
2. Computing the marginal contribution of each point to the model's performance.
3. Averaging the contributions across 10 permutations.

4.2 Results

The distribution of Shapley values is shown below:



4.3 Analysis

- Most Shapley values are close to zero, indicating that the majority of training points have minimal influence on the model's performance.
- A few points have higher Shapley values, suggesting they are more influential.

Conclusion

This assignment explored the influence of training data on model performance using LOO influence, group-level influence, and Shapley values. Key findings include:

1. **LOO Influence**: Most individual points have small negative influence, indicating they contribute positively to the model's performance.
2. **Group-Level Influence**: Larger groups tend to have higher influence, and some groups contain noisy or redundant data.
3. **Shapley Values**: Most points have minimal influence, but a few are highly influential.

These insights can help identify critical data points and improve model performance through targeted data curation.

AI Usage

Generative AI was used in this assignment to assist with various tasks, including code implementation, debugging, and report drafting. Below is a summary of how AI was utilized:

Code Implementation:

- AI provided code snippets for some of the key components of the assignment, such as data preprocessing, model training, and influence computation (e.g., LOO influence, group-level influence, Shapley values).
- These code snippets were adapted and customized to fit the specific requirements of the assignment.

Debugging and Optimization:

- AI was used to help identify and resolve errors in the code, such as handling missing data and improving model convergence.
- Suggestions for reducing runtime for Shapley value computation were also provided by AI and implemented after verification.

Report Writing:

- I wrote most of the report myself, including the introduction, methodology, and analysis sections. AI was used to improve clarity, structure, and grammar in certain parts of the report.
- For example, AI helped refine the wording of technical explanations and ensured the report was concise and aligned with the assignment requirements.

AI was used to assist during the assignment, with all AI-generated content carefully reviewed, modified, and validated to ensure it was correct and aligned with the assignment requirements. The final submission reflects my own understanding and effort, with AI serving as a tool to enhance productivity and accuracy.
