ULB UNIVERSITÉ LIBRE DE BRUXELLES

Project on Responsible Data Science

INFO-H420

Management of Data Science and Business Workflows

Students :

DESOIL Théo

ESIEV Kerim

PREVOST Louis

SEWIF Mohamed

TALHAOUI Youssef

VASILE George

Teacher:

SACHARIDIS Dimitris

GitHub Repository:

`https://github.com/Teytey2002/Project_Management.git`

2024-2025

Contents

Introduction

This project applies principles of responsible data science to analyze a machine learning pipeline for predicting income levels using the Adult dataset. The study addresses key aspects such as fairness, privacy, and explainability, ensuring that the model is both effective and ethical.

Key objectives include constructing and evaluating a classifier, improving fairness for protected attributes (e.g., age and sex), applying differential privacy to sensitive data, and analyzing the trade-offs between fairness, privacy, and model performance. Additionally, explainability techniques are explored to make the model's predictions more interpretable and accessible.

By integrating these principles, the project demonstrates how to build responsible machine learning systems while addressing critical challenges in fairness, privacy, and transparency.

Classification

The first stage of the project focuses on creating a reliable classifier for the Adult dataset. This step establishes the foundation for subsequent analysis by ensuring the model is trained and evaluated effectively. This process was divided into the following five key steps:

2.1 Data Preprocessing

- **Binarization of Age:** The *Age* column was binarized using its median value. This step was undertaken to divide individuals into two groups (*younger* and *older*), simplifying age representation in the model.
- **Handling Missing Values:** Rows with missing values were removed from the dataset. This choice ensured a clean dataset, avoiding potential biases introduced by imputation methods.
- **Feature Encoding:** Categorical variables were one-hot encoded to convert non-numerical data into a numeric format suitable for machine learning algorithms. This approach avoids introducing unintended ordinal relationships between categories.

2.2 Data Splitting

The dataset was split into three subsets:

- **Training Set:** Used to train the model.
- **Validation Set:** Used for hyperparameter tuning and model evaluation during training.
- **Test Set:** Used for final model evaluation to assess generalization to unseen data.

The data was divided in a ratio of approximately 60:20:20 to balance training and evaluation needs.

2.3 Model Selection

- **Choice of Classifier:** XGBoost was chosen as the classifier. This decision was guided by the dataset documentation, which highlighted its strong performance on structured data. XGBoost's ability to handle categorical features and optimize computational efficiency makes it particularly suitable for this task.
- **Pipeline Design:** A machine learning pipeline was implemented to standardize preprocessing and modeling. This ensured that the same transformations applied to the training data were consistently applied to validation and test datasets, minimizing risks of data leakage.

2.4 Model Evaluation

The model was evaluated using the following metrics:

- **Accuracy:** The proportion of correct predictions among all predictions.
- **F1 Score:** A measure that balances precision and recall, particularly important for datasets with class imbalances.
- **Confusion Matrix:** A visualization to assess the distribution of true positives, true negatives, false positives, and false negatives.

2.5 Visualization

A heatmap of the confusion matrix was generated to provide a detailed understanding of the model's predictions across the two income groups ($<=50K$ and $>50K$). This helped identify areas of improvement by visually interpreting the classification performance.

Fairness

The objective of this section is to ensure fairness in the classifier developed in the previous part. The process involves the following five steps:

3.1 Load Dataset and Specify Protected Attributes

The `AdultDataset` class, provided by the `aif360` library, is utilized to load and preprocess the Adult dataset. This dataset is widely used for studying fairness and bias in machine learning models.

A custom preprocessing function is defined to calculate the dataset's median age (37) and binarize the `age` attribute. Individuals aged above the median are assigned a value of `1` (older), while those below are assigned a value of `0` (younger). This preprocessing function is applied when calling the `AdultDataset` class.

The class also identifies the protected attributes (`age` (binarized), `sex`, and `race`) and the privileged group, defined as *older white males.* Even though the `race` attribute is predefined as a protected attribute by default in the `AdultDataset` class, we explicitly redefine the protected attributes to focus only on `age_binary` and `sex` as mentioned in the statement.

3.2 Compute Fairness Metric

The fairness metric used is the **Statistical Parity Difference (SPD)**, which quantifies the disparity in positive outcomes between the unprivileged and privileged groups. SPD is computed as:

$$SPD = P(\hat{y} = 1 \mid \text{Unprivileged Group}) - P(\hat{y} = 1 \mid \text{Privileged Group}), \tag{3.1}$$

where $\hat{y} = 1$ indicates a positive outcome.

In this case, the calculated SPD is **-0.363363**. This value indicates the presence of bias in the dataset, as an unbiased dataset would have an SPD equal to zero. Specifically, approximately **36% of individuals in the unprivileged group** experience unfair treatment.

3.3 Eliminate the Bias

To address the bias, the **Reweighting (RW)** pre-processing method is employed. This technique assigns weights to samples to balance the representation of protected groups during training.

The process involves creating an instance of the Reweighing class, followed by applying the `fit_transform` method to the original dataset (`dataset_orig`). This method executes the following steps:

- **Fit**: Computes weights for rebalancing the dataset.
- **Transform**: Applies the computed weights to the dataset.

3.4 Compare Classification Performance of Initial and Fair Models

To compare the performance of the fair model with the initial classifier:

1. Split the transformed dataset into training and testing sets using the `train_test_split` function.
2. Train and evaluate the **XGBoost Classifier** on the respective sets.

The results of the comparison are as follows:

- **Original Classifier:** The initial classifier achieved an accuracy of 0.872 and an F1 score of 0.92.
- **Fair Classifier:** After applying fairness mitigation techniques, the fair classifier achieved an accuracy of 0.864 and an F1 score of 0.91.

The results indicate that the accuracy of the model slightly decreases when fairness mitigation is applied. This decrease is expected, as fairness techniques often modify the decision boundaries of the model to ensure equitable outcomes across groups. This adjustment can lead to a minor loss in predictive performance, as the model prioritizes fairness over optimizing pure accuracy.

Privacy

The aim of this part is to compute the cross-tabulation of the sensitive attributes and do the same after applying Local Differential Privacy (LDP). Additionally, a classifier will be trained on the private dataset created through LDP, referred to as the **private classifier**, to evaluate the impact of privacy mechanisms on model performance.

4.1 Cross-Tabulation and Privacy Mechanism

A cross-tabulation of the sensitive attributes (*Age* and *Sex*) was computed to analyze the distribution of individuals across different combinations of these attributes.

LDP application

Local Differential Privacy (LDP) was applied to protect individual privacy by adding controlled noise to sensitive attributes (Age and Sex) at the individual level using randomized response mechanisms. The privacy level was managed using the parameter ϵ, balancing privacy and data accuracy. This process generated a new dataset with protected sensitive attributes.

After applying local differential privacy to the dataset, a new cross-tabulation was computed on the private data. The results showed the following:

- For $\epsilon = 0.1$, the noise introduced significantly altered the counts in the cross-tabulation. The error between the original and private cross-tabulations was quantified using the Mean Absolute Error (MAE), which was calculated as **8.21%**.
- For $\epsilon = 0.5$, the noise also significantly altered the counts, but the Mean Absolute Error (MAE) was calculated to be **6.68%**, slightly lower than $\epsilon = 0.1$. This indicates that while noise is still present, its impact on the data is less pronounced with a higher epsilon value.

- For $\epsilon = 1.0$, the private cross-tabulation more closely approximated the original counts, with an MAE of **4.91%**.
- For $\epsilon = 2.0$, the private dataset provided a high level of accuracy, with an MAE of **2.04%**, although this resulted in reduced privacy guarantees.

These results demonstrated the trade-off between privacy and accuracy: lower ϵ values provided stronger privacy guarantees at the cost of higher errors, while higher ϵ values improved accuracy but reduced privacy. So We used $\epsilon = 1.0$ for the rest of the privacy part due to its compromise between accuracy and privacy.

Error Estimation in Cross-Tabulations

To quantify the impact of applying local differential privacy on sensitive attributes (*Age* and *Sex*), the following steps were performed:

- The corrected estimations for the total counts of *Sex* and *Age* were computed based on the private dataset:
  - Estimated total for *Sex*: **30,937.0**.
  - Estimated total for *Age*: **22,055.0**.
- The total errors in estimation were calculated by comparing the private cross-tabulation to the original data:
  - Error for class 0: **1,848**.
  - Error for class 1: **-1,848**.

These results indicate that the noise introduced by the local differential privacy mechanism causes significant variations in the estimated totals. The positive and negative errors suggest that the mechanism redistributes the data across sensitive attribute categories, maintaining privacy at the expense of accuracy.

Discussion of Errors

The errors in estimation highlight the inherent trade-offs in using local differential privacy:

- Stronger privacy guarantees (lower ϵ) lead to larger estimation errors, reducing the fidelity of the private data.
- The corrections applied to the private data are effective in approximating the original totals but cannot completely eliminate discrepancies introduced by the privacy mechanism.

- The symmetric nature of the errors (e.g., equal but opposite errors for classes 0 and 1) reflects the randomized nature of the mechanism, which ensures fairness but sacrifices precision.

4.2 Private classifier

The private dataset was split into training and testing sets in the same manner as the original dataset. A machine learning classifier (XGBoost) was trained on the private data, and its performance was measured using the following metrics:

- **Accuracy**: The overall accuracy of the private classifier was **87.24%**, demonstrating a reasonable performance despite the application of local differential privacy.
- **Precision**: The precision for class 0 (income $\leqslant 50K$) was **0.90**, while for class 1 (income $> 50K$), it was **0.78**. This indicates that the model is more confident in its predictions for individuals earning less than $50K$.
- **Recall**: The recall for class 0 was **0.94**, whereas for class 1, it was **0.67**. This suggests that the model is better at identifying individuals earning less than $50K$ compared to those earning more.
- **F1-Score**: The F1-score, which balances precision and recall, was **0.92** for class 0 and **0.72** for class 1, resulting in a weighted average F1-score of **0.87**.

These metrics highlight that the classifier trained on the private dataset performs well overall, but there is a noticeable drop in recall and F1-score for individuals with income $> 50K$. This can be attributed to the noise introduced by the local differential privacy mechanism, which may obscure patterns in the sensitive data.

Impact of Privacy on Model Performance

Comparing the private classifier's performance with the original classifier trained on non-private data reveals the following:

- The classifier's overall accuracy remained relatively high at **87.24%**, with only a slight reduction compared to the original classifier (assumed to be close to 90%).
- The performance drop was more pronounced for class 1 (income $> 50K$), particularly in terms of recall and F1-score, due to the added noise in the sensitive attributes.
- These results illustrate the trade-off between privacy and model performance, emphasizing the importance of selecting an appropriate level of privacy (parameter ϵ) to balance these competing goals.

Privacy and Fairness

In this section, we aim to create a classifier that is both private and fair, referred to as the private and fair classifier. We then evaluate its fairness and compare it with the fairness of the classifier designed to address fairness concerns (fair classifier). This involves integrating principles from differential privacy and fairness mitigation techniques.

5.1 Private and Fair Classifier

To construct the private and fair classifier, we start with the fair classifier developed in Section 2. We then apply the ϵ-differential privacy mechanism, which ensures privacy by adding controlled noise to the data. The parameter ϵ quantifies privacy loss, with smaller values indicating stronger privacy guarantees. This step modifies the fair classifier to satisfy privacy requirements while preserving its fairness-enhancing mechanisms.

5.2 Comparaison

To evaluate the impact of privacy and fairness mechanisms, we compute and compare the fairness metrics across various datasets and stages:

- **Baseline Dataset (Pre-reweighting):** The fairness metric is 36%. This serves as the starting point without any fairness mitigation or privacy mechanisms applied.
- **Private Dataset (Pre-reweighting):** The fairness metric drops to 18% after applying differential privacy. This decrease occurs because the noise introduced by the differential privacy mechanism distorts the sensitive attributes (e.g., Age, Sex) and their relationship to the target variable. This makes it more challenging to assess and mitigate unfairness effectively.
- **Reweighted Fair Classifier:** After applying the fairness reweighting method, the fairness metric is reduced to 0%. This demonstrates the effectiveness of the reweighting

approach in eliminating unfairness.

- **Reweighted Private and Fair Classifier:** Similarly, the fairness metric for the private and fair classifier is also 0% after reweighting. This shows that despite the noise introduced by differential privacy, the fairness mitigation technique can still achieve equitable outcomes.

Explainability

This section explores the explainability of the private classifier trained on the Adult dataset. The study identifies instances where the model is highly confident but makes incorrect predictions. Additionally, it investigates whether noisy sensitive attributes, such as `age_binary_private` and `sex_binary_private`, contribute to the model's overconfidence and errors.

6.1 Global Explainability

To analyze the overall behavior of the model, we employed two global explainability techniques:

- **Accumulated Local Effects (ALE):** This method visualizes how features influence the model's predictions on average, highlighting interactions and nonlinear relationships.
- **SHAP (SHapley Additive exPlanations):** This approach quantifies the contribution of each feature to the predictions, offering insights into feature importance.

The results revealed that features such as `Capital Gain`, `Workclass`, and `Education` had significant influence on the model's decisions. The noisy sensitive attributes, while important, displayed patterns that might contribute to prediction distortions.

6.2 Identifying Highly Confident but Incorrect Predictions

We identified instances where the model made incorrect predictions with high confidence. The process involved:

1. Finding instances where the predicted label differed from the true label.
2. Filtering these instances to retain only those with confidence scores exceeding 95%.

A total of **{len(miss_but_confident)}** instances were found to meet these criteria. For example:

- **True label:** $> 50K$.
- **Predicted label:** $\leqslant 50K$.
- **Confidence:** 97.4%.

6.3 Local Explainability

For each identified instance, we used the following techniques to explain the model's decisions:

- **LIME (Local Interpretable Model-agnostic Explanations):** LIME approximates the model locally with a linear model, identifying features that contributed most to the prediction.
- **MACE (Model-agnostic Counterfactual Explanations):** MACE generates counterfactuals—alternative inputs that change the prediction—to explore minimal changes needed to alter the model's decision.

For instance, LIME highlighted features such as `Capital Gain` and `Marital Status` as key contributors to an incorrect prediction. MACE revealed that small adjustments to `Capital Gain` could correct the model's decision.

6.4 Investigating the Impact of Noisy Sensitive Attributes

Given access to the real values of `Age` and `Sex`, we evaluated whether their noisy counterparts (`age_binary_private` and `sex_binary_private`) were responsible for the model's overconfidence and errors. This involved:

1. Comparing the factual data (with noisy sensitive attributes) against counterfactual data (with real sensitive attributes).
2. Predicting outcomes for both factual and counterfactual data.

For an example instance:

- **Factual Prediction:** $\leqslant 50K$ with confidence 97.4%.
- **Counterfactual Prediction:** $> 50K$ with confidence 89.2%.

These results demonstrate that the noise introduced in sensitive attributes, such as `age_binary_private` and `sex_binary_private`, significantly impacted the model's predictions. Correcting these attributes improved accuracy and reduced overconfidence.

Explainability and LLMs

7.1 Identifying Highly Confident but Incorrect Predictions

The code used to identify instances where the model is highly confident but makes incorrect predictions was adapted from Part 5 of this project. While the original implementation included multiple explainers, we focused exclusively on **LIME** for our analysis in this section.

1. **Finding Incorrect Predictions:** Instances where the predicted label differed from the actual label were identified as follows:

   ```
   miss_indices = np.where(predictions != test_labels)[0]
   ```

2. **Filtering for High Confidence:** From the incorrect predictions, we retained those where the model's confidence exceeded 95%:

   ```
   if max(proba[idx]) > 0.95:
       miss_but_confident.append(idx)
   ```

7.2 Explaining Predictions Using LIME

We applied the LIME explainer to the identified instances to understand the key factors influencing the model's predictions.

We selected a subset of instances for visualization and generated local explanations. For each instance, the LIME explainer provided insights into the key features and their contributions.

7.2.1 Translating Explanations to Text

The feature importance scores from LIME were extracted and converted into a readable textual format. An example explanation is shown below:

```
For this individual, the key factors are:
- Capital Gain: contribution of 0.39
- Marital Status: contribution of -0.23
- Capital Loss: contribution of 0.20
- Education: contribution of 0.16
- Age: contribution of -0.06
- Sex: contribution of -0.04
- Country: contribution of 0.03
```

7.3 Using LLM for Explainability

To enhance the readability of the explanations, we used the application **LM Studio** running the **Llama 3.2 3B Instruct** model locally on our machine. The generated textual explanations were sent to the LLM for reformulation into more human-readable insights. For instance, the above explanation was transformed into:

"Capital Gain: For every dollar earned in capital gains (like from investments), they get 39 cents more. Marital Status: Getting married actually hurts them - it gives them a 23-cent penalty. Education: Having a higher education level helps them out - it gives them a 16-cent boost."

The interaction process with the LLM ensures that explanations are presented in an intuitive and user-friendly format.

Conclusion

This project demonstrates how principles of fairness, privacy, and explainability can be integrated into a machine learning pipeline. Using the Adult dataset, we built a classifier that not only achieves high performance but also ensures ethical considerations are met.

Fairness was addressed by mitigating biases in sensitive attributes such as age and sex, improving equitable outcomes. Privacy was safeguarded using local differential privacy, which protected sensitive data while maintaining acceptable utility. Explainability techniques, including LIME and counterfactual explanations, provided insights into the model's decisions, especially for confident yet incorrect predictions. These explanations were further enhanced using the **Llama 3.2 3B Instruct** model via **LM Studio**, translating technical results into user-friendly narratives.

While trade-offs between fairness, privacy, and performance were evident, this project highlights the feasibility of developing responsible AI systems. The approach offers a framework for building machine learning models that are accurate, equitable, and transparent, paving the way for trustworthy AI in real-world applications.