ECOLE POLYTECHNIQUE DE BRUXELLES

Management of Data Science and Business Workflows

Project on Responsible Data Science

Kristóf Balázs

Nishant Sushmakar

Viet Phuong Hoang

Nhu Ngoc Hoang

Ngoc Hoa Pham
Olha Baliasina

Prof. Dimitris SACHARIDIS

2024

Contents

Introduction

The goal of this project is to combine principles from responsible data science to study an ML pipeline. Specifically, this project is focused on studying a classifier that predicts whether someone will have income “>50K” on the Adult data set. The project aims to investigate several responsible data science aspects of the ML pipeline, such as:

1. Fairness
2. Privacy
3. Explainability
4. Application of LLMs

All the stages of this project, completed by our team, will be presented in the following chapters of this report.

1 Classification

Task description:

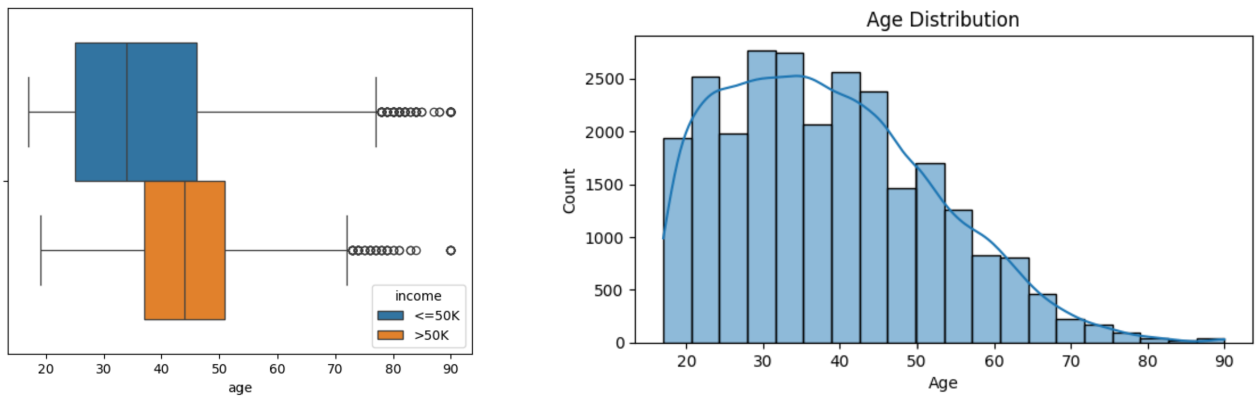
> Preprocess the data, and binarize Age. Split the data into train, validation, test sets, and train a classifier; we will refer to it as the classifier. Measure the performance of the classifier on the test set.

1.1 Initial Exploration and Setup

Before developing a custom pipeline, we used initial model exploration and comparison. The automated machine learning (AutoML) exploration can be found in the `automl_adult_analysis.ipynb` notebook. The setup handled preprocessing (e.g., missing values, scaling, encoding) and applied stratified K-Fold cross-validation. Using `compare_models`, we evaluated multiple classifiers. Based on the results, we used the top-performing models based on F1 score and Area Under the Curve (AUC) in model comparison with our preprocessing pipeline. The selected models were the following (in order): Light Gradient Boosting Machine (LightGBM), Extreme Gradient Boosting (XGBoost), Gradient Boosting Classifier, Ada Boost Classifier, Logistic Regression, Random Forest Classifier, Linear Discriminant Analysis (LDA), Ridge Classifier, K Neighbors Classifier (KNN), Extra Trees Classifier, Quadratic Discriminant Analysis (QDA), Naive Bayes, SVM - Linear Kernel, and Dummy Classifier.

1.2 Data Preprocessing

We identified missing values in the dataset, mainly in the `workclass`, `occupation`, and `native_country` columns. We decided to handle these using the `SimpleImputer` with the `most_frequent` strategy. Whitespaces were also removed from string columns. Based on the distribution of the data visible in Figure 1.1, we binarized the age feature based on age **25**. Finally, we label-encoded the target variable `income`.



(a) Boxplot of Age by Income Group

(b) Age Distribution Histogram

Figure 1.1: Visualization of Age and Income Data

We first split the dataset into a combined training/validation set (80%) and a test set (20%) using Stratified K-Fold Cross-Validation. We then further divided the training/validation set into separate training (75%) and validation (25%) sets using another round of Stratified K-Fold Cross-Validation.

Our **preprocessing pipeline** fills missing values in numerical columns with the median and scales them. For categorical columns, we use one-hot encoding to convert them into a numerical format that the models can interpret. Both are applied through a `ColumnTransformer`, which processes numerical and categorical columns in parallel. The final pipeline includes a step to convert sparse matrices (from one-hot encoding). We did this to make sure that this pipeline is compatible with the models we selected for comparison.

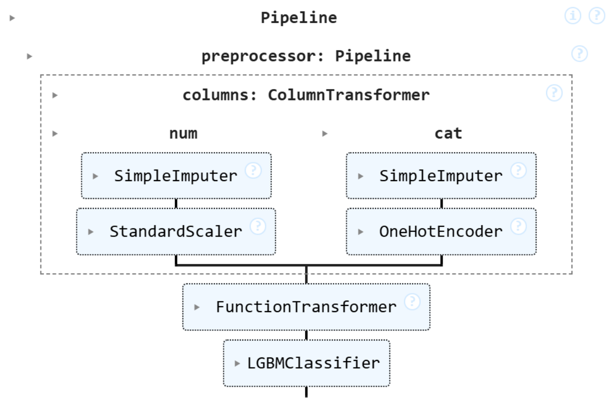
1.3 Model Training and Evaluation

Using our preprocessing pipeline, We evaluated several models using a 5-fold stratified cross-validation. The results are visible in Table 1.1 below.

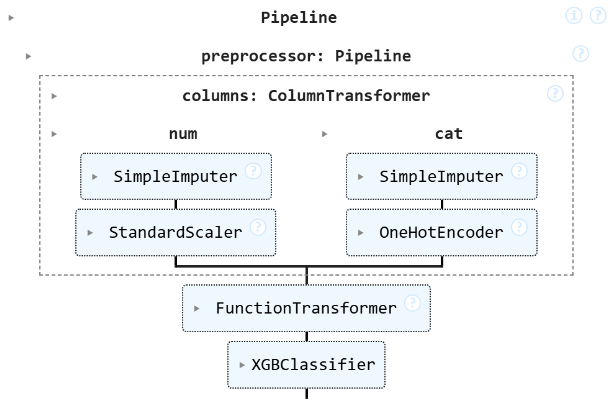
Model	F1-Weighted	Accuracy	AUC	Recall	Precision
XGBoost	0.860581	0.864942	0.919966	0.636800	0.763207
LightGBM	0.860388	0.865211	0.920410	0.629626	0.768802
Gradient Boosting	0.854703	0.861564	0.914159	0.593116	0.779376
AdaBoost	0.849242	0.855959	0.909230	0.587376	0.759966
Logistic Regression	0.843311	0.849163	0.902750	0.589768	0.731752
SVM	0.843250	0.851083	0.890977	0.565373	0.754594
Random Forest	0.832070	0.835112	0.878607	0.607783	0.675575
LDA	0.831298	0.838798	0.888546	0.552141	0.713578
Ridge Classifier	0.824975	0.837109	0.888581	0.495858	0.742102
KNN	0.824188	0.828164	0.849416	0.581957	0.663086
Extra Trees	0.811448	0.813345	0.837137	0.584026	0.619220
Dummy	0.655299	0.759214	0.500000	0.000000	0.000000
Naive Bayes	0.548757	0.533630	0.737217	0.953445	0.335555
QDA	0.338516	0.383296	0.805862	0.950409	0.279883

Table 1.1: Model Evaluation After Binarization

The two best performances were by `LightGBM` and `XGBoost`. We chose `LightGBM` due to slightly better precision (76.9% vs. 76.3%) and F1-Weighted (0.8604 vs. 0.8606). We trained both `LightGBM` and `XGBoost` on the combined training and validation sets and evaluated it on the test set. The `LightGBM` classifier will be used in the following chapter, that is **the classifier**. On Figure 1.2, both pipleines are visible.



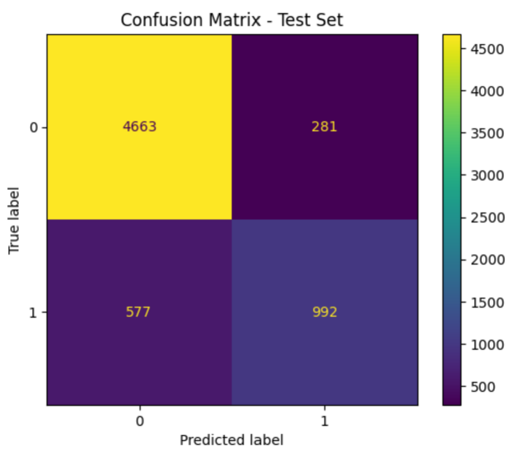
(a) Pipeline with LGBMClassifier



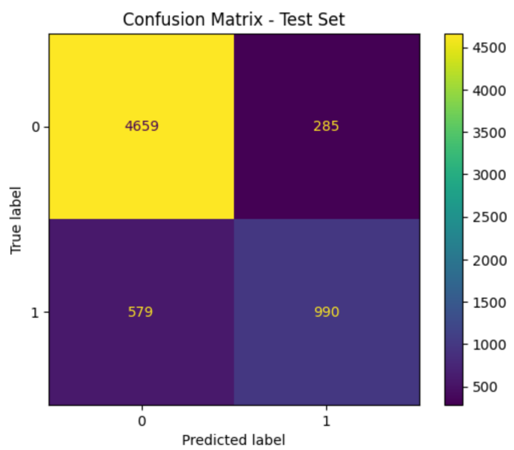
(b) Pipeline with XGBClassifier

Figure 1.2: Comparison of Pipelines with Different Classifiers

We plotted the confusion matrix for the test set for both models, which shows that they performs well overall, with relatively few false positives and false negatives. Both matrices are visible on Figure 1.3.



(a) Confusion Matrix - LGBMClassifier



(b) Confusion Matrix - XGBClassifier

Figure 1.3: Comparison of Confusion Matrices for LGBMClassifier and XGBClassifier

Lastly, we measured the performance of both classifiers on the test set. The results are visible on Table 1.2. As we mentioned before, we chose the `LightGBM` classifier, which we will later refer to as **the classifier**.

(a) LGBMClassifier - Classification Report

Metric	Values
Precision (0, 1)	0.89, 0.78
Recall (0, 1)	0.94, 0.63
F1-Score (0, 1)	0.92, 0.70
Support (0, 1)	4944, 1569
Accuracy	0.86826
Macro Avg (P, R, F)	0.83, 0.79, 0.81
Weighted Avg (P, R, F)	0.86, 0.87, 0.86

(b) XGBClassifier - Classification Report

Metric	Values
Precision (0, 1)	0.89, 0.78
Recall (0, 1)	0.94, 0.63
F1-Score (0, 1)	0.92, 0.70
Support (0, 1)	4944, 1569
Accuracy	0.86734
Macro Avg (P, R, F)	0.83, 0.79, 0.81
Weighted Avg (P, R, F)	0.86, 0.87, 0.86

Table 1.2: Classification Reports for XGBClassifier and LGBMClassifier

2 Fairness

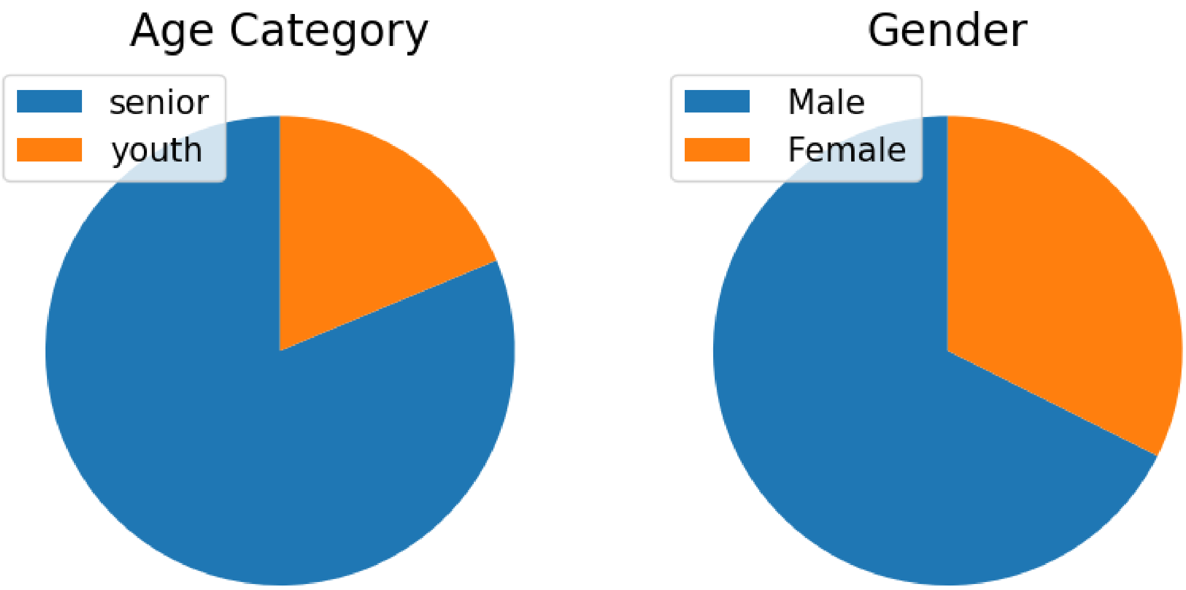
Task description:

> Assess the group fairness of the classifier, assuming the protected attributes are Age, Sex. Choose any fairness metric, and apply a technique to ensure the classifier is fair. We will refer to it as the fair classifier. Report the chosen fairness metric on the classifier and on the fair classifier.

2.1 Fairness Exploration

Datasets can have biases that reflect unfair patterns in society. These biases can cause machine learning models to make unfair decisions, favoring one group over another. It is important to find and fix these biases to make AI systems more fair and reliable.

We did an initial exploratory analysis on the data to check the level of bias in the dataset. The binarization of age was set to 25 and people above 25 years are categorised as seniors and otherwise they are youth. We also assume that based on societal norms the underprivileged class are the youth in the age category and the females in gender category. In figure 2.1 given below we can clearly see that the privileged classes dominate the datasets. If a privileged class dominates a dataset, the model is likely to learn and reinforce biases, leading to unfair outcomes for underrepresented groups.



(a) Pie Plot on Age Distribution

(b) Pie Plot on Gender Distribution

Figure 2.1: Visualization of Age and Gender Distribution in Data

We have encoded incomes greater than $50K as 1 and incomes less than or equal to $50K as 0. This implies that a model predicting 1 indicates a potential benefit for the individual. For example, if a bank uses these predictions to offer loans, individuals with a predicted income of 1 (i.e., >$50K) would likely qualify for a loan, thereby gaining an advantage from the model's predictions.

To check this benefit in this model we calculate prevalence which is the proportion of the positive cases to overall cases. Where a positive case is when the target variable has a value of 1.

Age Category	**Prevalence(%)**
Youth	1.927764
Senior	30.171755

Table 2.1: Age Prevalence

Sex Category	**Prevalence(%)**
Female	11.167967
Male	31.464117

Table 2.2: Sex Prevalence

Age Category	**Sex Category**	
	Female	**Male**
Youth	1.283368	2.417154
Senior	14.448799	36.881908

Table 2.3: Age and Sex Combination Prevalence

The tables show that in each group, the privileged class has a much higher prevalence. For example, seniors have a prevalence of 30%, and males have 31%. Senior males have the highest prevalence at 36%. This dataset, based on 1994 United States Census data, highlights the lack of opportunities for young people to earn high incomes and the presence of gender discrimination.

2.2 Classifier Fairness Assessment

The best model from the experiments was the LGBM classifier, which uses the gradient boosting framework. This method builds models sequentially, each one correcting the mistakes of the previous one. The model was selected based on the average validation F1-macro score, and the best one was chosen based on the highest score achieved in a fold.

Class	Precision	Recall	F1-Score	Support
<=$50K	0.89	0.94	0.92	4527
>$50K	0.79	0.66	0.72	1501
Accuracy	0.87			6028
Macro avg	0.84	0.80	0.82	6028
Weighted avg	0.87	0.87	0.87	6028

Table 2.4: Classification Report on Test Dataset with Classifier

The model performance is given in Table 2.4 where you can see that f1-macro is 0.82 on the test set and the respective metric on each classes.

To assess the fairness of the classifier, we calculate the True Positive Rate (TPR) ratio between the underprivileged and privileged groups. The True Positive Rate represents the percentage of individuals from each group who were correctly identified by the model as benefiting from the outcome. If the TPR ratio is equal or near to 1, it indicates perfect fairness, meaning both groups have an equal chance of benefiting from the model.

Group	True Positive Rate	False Positive Rate	False Negative Rate	Disparate Impact
Senior	0.664861	0.078916	0.335139	0.256515
Youth	0.375000	0.001768	0.625000	0.009524
Ratio	0.564027	0.022408	1.864899	0.037128

Table 2.5: Fairness Metrics for Age Category

Group	True Positive Rate	False Positive Rate	False Negative Rate	Disparate Impact
Male	0.669572	0.085327	0.330428	0.266798
Female	0.610879	0.017961	0.389121	0.090076
Ratio	0.912342	0.210492	1.177629	0.337620

Table 2.6: Fairness Metrics for Sex Category

Group	True Positive Rate	False Positive Rate	False Negative Rate	Disparate Impact
Senior Male	0.673633	0.110083	0.326367	0.315913
Young Female	0.333333	0.002033	0.666667	0.006024
Ratio	0.494829	0.018463	2.042693	0.019069

Table 2.7: Fairness Metrics for Senior Male and Young Female Groups

The table presents various fairness metrics, but for our assessment, we focus on the True Positive Rates (TPR). As seen, the ratio of TPR between youth and seniors is approximately 0.56, indicating that seniors benefit much more from the model than the youth. A similar pattern is observed in the TPR ratio between females and males, with females having a lower rate. However, the disparity becomes even more pronounced when comparing the most prevalent group (senior males) to the least prevalent group (young females), where the TPR ratio is about 0.49. This significant difference highlights that the classifier is unfair, favoring the more privileged groups while disadvantaging the underprivileged class.

2.3 Ensuring Fairness in Classifier Design

There are several fairness methods to adjust classifiers, including pre-processing, in-processing, and post-processing changes. Pre-processing is the most effective for fairness because it tackles bias in the data before training, helping the model learn from a more balanced dataset. Unlike post-processing and in-processing, it works independently of the model and can be used with different algorithms, making it more flexible and scalable. In our approach, we use the Reweighing algorithm to adjust the dataset weights during training.

The reweighing process works as follows: We start with a dataset that is divided into a main dataset and a test dataset. The main dataset is split into 5 folds using stratified K-fold cross-validation. In each iteration, one fold is selected for validation, and the remaining folds are used for training. The training data is then passed to the reweighing algorithm, where we specify the underprivileged and privileged groups. The algorithm calculates new weights, which are then used during model training to adjust for fairness.

2.4 Fair Classifier Fairness Assessment

After applying the reweighing process, the model is selected based on the best validation score. Once the best model is chosen, it is tested on the initially split test dataset to evaluate its performance. During training the validation score dropped from 0.808 to 0.78, indicating a degradation in the classifier's performance after applying the fairness adjustments.

Class	**Precision**	**Recall**	**F1-Score**	**Support**
<=$50K	0.87	0.96	0.91	4527
>$50K	0.83	0.55	0.66	1501
Accuracy		0.86		6028
Macro Avg	0.85	0.76	0.79	6028
Weighted Avg	0.86	0.86	0.85	6028

Table 2.8: Classification Report on Test Dataset with Fair Classifier

We also observe a noticeable degradation in the test dataset classification report, where the F1-macro score drops from 0.82 to 0.79. Additionally, the model's performance decreases for each class, indicating that the reweighing process has led to a reduction in overall effectiveness.

As we can see in the given below table, the True Positive Rate has significantly improved for both the age and sex categories. The most underprivileged group is now benefiting more from the model, with the True Positive Rate increasing from 0.49 to 1.55, showing a major improvement in the fairness of the classifier. However, this comes with a tradeoff, as the overall performance of the model has degraded.

Group	True Positive Rate	False Positive Rate	False Negative Rate	Disparate Impact
Senior	0.553825	0.047703	0.446175	0.201108
Youth	0.500000	0.009726	0.500000	0.019913
Ratio	0.902812	0.203884	1.120637	0.099018

Table 2.9: Fairness Metrics for Age Category with Fair Classifier

Group	True Positive Rate	False Positive Rate	False Negative Rate	Disparate Impact
Male	0.534073	0.045341	0.465927	0.197145
Female	0.652720	0.026651	0.347280	0.102799
Ratio	1.222155	0.587796	0.745353	0.521439

Table 2.10: Fairness Metrics for Sex Category with Fair Classifier

Group	True Positive Rate	False Positive Rate	False Negative Rate	Disparate Impact
Senior Male	0.536174	0.057817	0.463826	0.232531
Young Female	0.833333	0.018293	0.166667	0.028112
Ratio	1.554223	0.316390	0.359330	0.120898

Table 2.11: Fairness Metrics for Senior Male and Young Female Groups with Fair classifier

We can now confidently say that the fair classifier lives up to its name, as it is effectively uplifting the underprivileged class.

3 Privacy

Task description:

> Assume that the attributes Age, Sex are sensitive. Compute a cross-tabulation showing how many people exist in value combinations of the two sensitive attributes. Apply local differential privacy to the responses of the individuals about what is their Age and Sex, and create a private data set. You may want to explore various epsilon values. Compute a cross-tabulation on the private data, and estimate how many people exist in value combinations of the two sensitive attributes. Quantify the errors in the estimation. Split the private data in the same manner as in (1), and train a classifier; we will refer to it as the private classifier. Measure the performance of the private classifier. Check if there is an impact on model performance due to privacy compared to the classifier.

3.1 Local Differential Privacy

In many machine learning scenarios, sensitive attributes can leak personal information or enable discrimination against certain groups. Privacy-preserving techniques aim to lower these risks. One such technique is **Local Differential Privacy** (LDP), where the data of each individual is perturbed before being collected. Unlike classical differential privacy (which is typically applied by a data curator), LDP ensures privacy at the data source level, meaning even the data collector cannot learn the exact original values.

By applying LDP to sensitive attributes, we ensure that individual responses about attributes are randomized in a controlled manner. The strength of privacy is governed by the parameter ϵ: smaller values of ϵ lead to stronger privacy at the cost of higher distortion in the data.

3.2 Exploratory analysis of sensitive attributes

We consider Age and Sex as sensitive attributes. Similar to the fairness setting, we encode Age into two categories, *Senior* (Age > 25) and *Young* (Age ≤ 25), and Sex as *Female* and *Male*. We create a cross-tabulation of these attributes to understand the distribution.

Age	Female	Male	All
Senior	7950	18200	26150
Young	2821	3590	6411
All	10771	21790	32561

Table 3.1: Cross-tabulation of Age and Sex in the original dataset

As shown in Table 3.1, the majority of individuals are senior males, while young females are the least represented. This pattern is consistent with previous observations of demographic distributions in the dataset.

When including the target variable in the cross-tabulation for exploratory data analysis purposes (Table 3.2), we observe that the senior male category represents approximately 6-7 out of every 8 individuals with an income exceeding 50K in the training dataset.

Age	Sex	Income <=50K	Income >50K	All
Senior	**Female**	6803	1147	7950
	Male	11620	6580	18200
Young	**Female**	2789	32	2821
	Male	3508	82	3590
All		24720	7841	32561

Table 3.2: Cross-tabulation of Age, Sex and Income in the original dataset

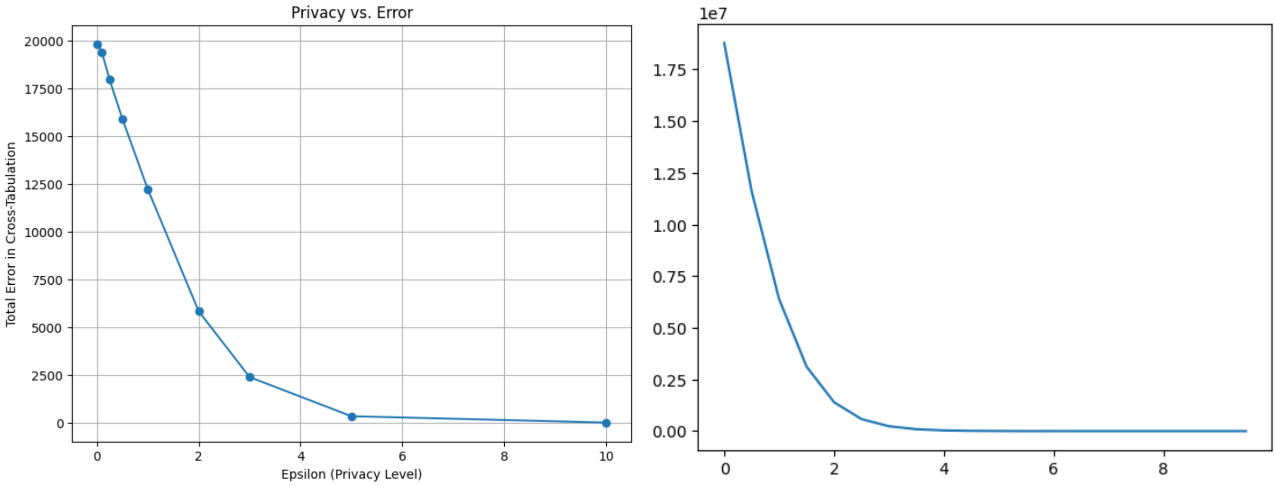
3.3 Applying Local Differential Privacy to the sensitive attributes

To protect the attributes Age and Sex, we apply a randomized response mechanism, one of the LDP techniques. For a given ϵ, each individual's sensitive attribute values are perturbed. Intuitively, for lower ϵ, the response is closer to a random guess, ensuring stronger privacy but introducing more noise. For larger ϵ, the mechanism is more faithful to the original data, reducing noise but also reducing privacy.

For our case, we assumed that $p = q$ and our implementation of randomized response takes ϵ as a parameter. We experimented with various ϵ values, ranging from very small values (0.01) to larger values (such as 10). After applying LDP, we computed new cross-tabulations on the perturbed attributes and quantified the total error compared to the original counts.

3.4 Impact of ϵ on estimation errors

We defined the *total error* as the sum of the absolute deviations in the cross-tabulation cells between the private (perturbed) data and the original data. In addition, we analyzed the variance of errors in multiple trials for different ϵ values. Figure 3.1 and Table 3.3 summarize the outcomes we obtained:



(a) Privacy vs. Error graph

(b) Variance estimation

Figure 3.1: Impact of ϵ on estimation errors

ϵ	0.01	0.1	0.25	0.5	1	2	3	5	10
Total Error	19830	19388	17956	15908	12220	5828	2384	336	4

Table 3.3: Total error in cross-tabulation estimates at various ϵ values

As shown, extremely low ϵ values introduce significant distortion, while larger ϵ values approach near-perfect recovery of original distribution. For instance, at $\epsilon = 10$, the total error is almost negligible. The results also show that the variance of the estimation error decrease significantly as ϵ increases. This once again confirms the fundamental privacy-utility trade-off: stronger privacy (smaller ϵ) leads to higher variance and greater estimation error, while weaker privacy (larger ϵ) means more stable and accurate estimates.

3.5 Creating a private dataset and a private classifier

Having selected an ϵ value (for demonstration and in order to ensure high level of privacy, $\epsilon = 0.5$), we used the perturbed attributes to create a *private dataset.* The results of cross-tabulation comparison between the original and private dataset are presented in the Table 3.4.

	Female	**Male**
Young	2821	3590
Senior	7950	18200

(a) Original cross-tabulation

	Female	**Male**
Young	6340	7498
Senior	8607	10116

(b) Private cross-tabulation

	Female	**Male**
Young	3519	3908
Senior	657	-8084

(c) Error Matrix

Table 3.4: Comparison of original and private cross-tabulations

The private dataset is then split into training and testing sets following the same procedure used in the original (non-private) scenario.

We train a LightGBM classifier (referred to as the *private classifier* further in the report) on the private dataset. The goal is to evaluate whether applying LDP to sensitive attributes affects predictive performance. Similar to the fairness setting, we applied stratified k-fold cross-validation for model selection.

Class	Precision	Recall	F1-Score	Support
$\leq 50K$	0.89	0.94	0.91	4527
$> 50K$	0.78	0.64	0.71	1501
Accuracy		0.87		6028
Macro avg	0.83	0.79	0.81	6028
Weighted avg	0.86	0.87	0.86	6028

Table 3.5: Classification report on the test dataset with the private classifier ($\epsilon = 0.5$)

Table 3.5 shows the private classifier's performance on the test set. While the results remain strong (accuracy around 0.87), there is a slight decrease compared to the non-private classifier (which had an F1-macro of about 0.82 originally, and now it is about 0.81). This suggests that adding noise to protect privacy does have a measurable, though not drastic, impact on predictive performance.

3.6 Assessing Fairness and the Impact of Privacy

We have also assessed fairness-related metrics on the private classifier. While the core objective here is privacy, it is worth noting that privacy-preserving transformations can sometimes affect fairness. In this case, the True Positive Rate (TPR) differences among groups persisted, but there was some improvement in parity for underrepresented groups. The given model is still unfair to Young and Female class, however, in comparison to the classifier, it is noticebly more fair. In particular, the True Positive Rate metric showed huge increase for the Young and Young Females underpriveleged classes.

Group	Accuracy	True Positive Rate	False Positive Rate	False Negative Rate	Disparate Impact
Senior	0.840345	0.646581	0.075383	0.353419	0.248512
Young	0.974892	0.541667	0.015915	0.458333	0.026840
Ratio	1.160109	0.837740	0.211124	1.296855	0.108002

Table 3.6: Fairness Metrics for Age Category with Fair Classifier

Group	Accuracy	True Positive Rate	False Positive Rate	False Negative Rate	Disparate Impact
Male	0.834851	0.653724	0.083542	0.346276	0.260645
Female	0.930789	0.598326	0.023175	0.401674	0.093130
Ratio	1.114916	0.915258	0.277406	1.159982	0.357305

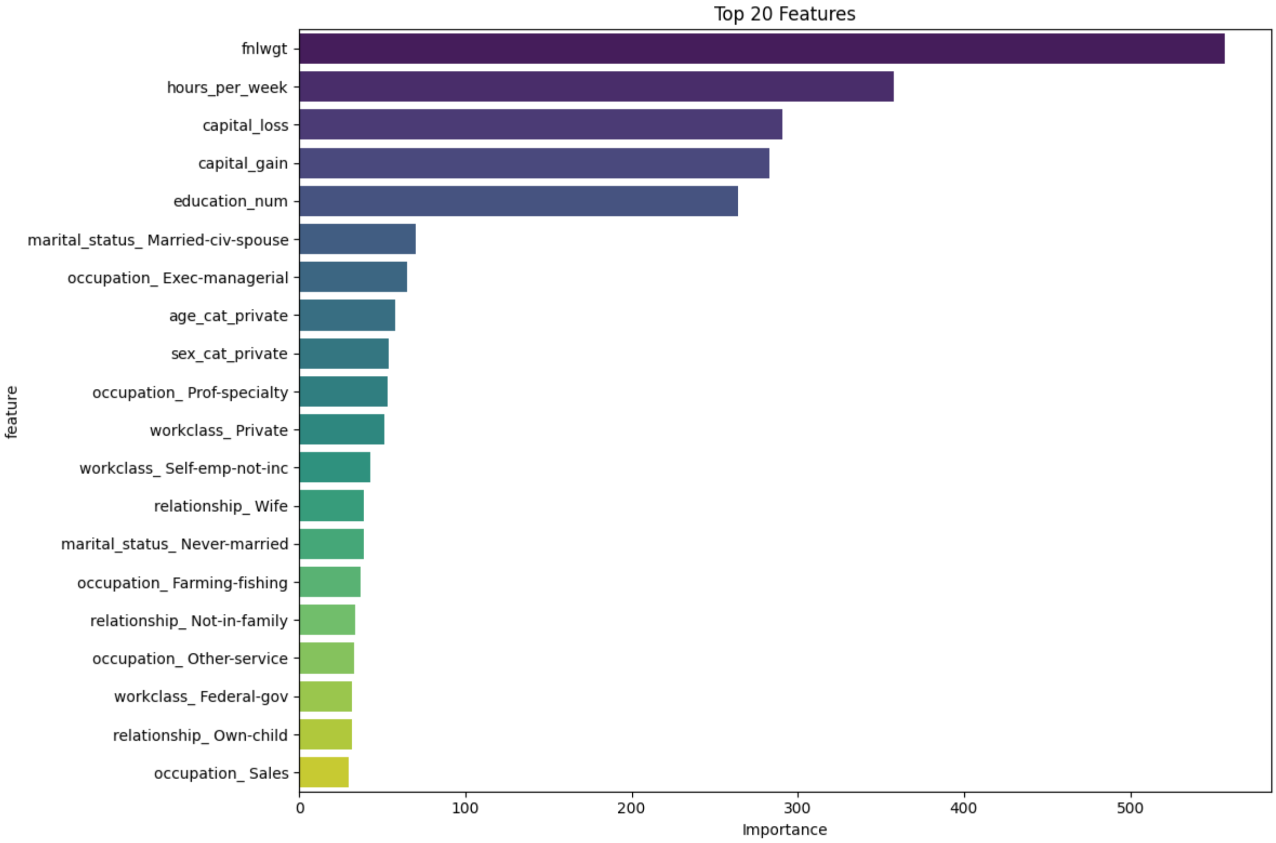
Table 3.7: Fairness Metrics for Sex Category with Fair Classifier

Group	Accuracy	True Positive Rate	False Positive Rate	False Negative	Disparate Impact
Senior Male	0.808867	0.655145	0.102683	0.344855	0.304463
Young Female	0.981928	0.500000	0.012195	0.500000	0.018072
Ratio	1.213955	0.763190	0.118765	1.449883	0.059358

Table 3.8: Fairness Metrics for Senior Male and Young Female Groups with Fair Classifier

3.7 Feature analysis

In order to try to find the explanation of the results obtained earlier we extracted top-20 most important features for the chosen model.



From the bar chart we can observe that the Sex and Age features are not of the highest importance for the target. That can explain the non-significant difference in performance of the classifier and the private classifier.

4 Privacy and Fairness

Task description:

> Consider the same fairness metric and fairness mitigation method as in (2). Create a fair version of the private classifier; we will refer to is as private+fair classifier. Assume, you're an auditor that has access to the real sensitive values of Age and Sex. Using the real values of Age and Sex, measure the fairness of the private+fair classifier, and compare it to that of the fair classifier. Draw conclusions.

Up to this point, we have explored fairness and privacy separately. In practice, both aspects often need to be addressed simultaneously. Introducing local differential privacy (LDP) ensures that sensitive attributes (such as Age and Sex) are protected at the source. However, this noise addition can distort data distributions and reduce the effectiveness of downstream fairness mitigation techniques.

Previously, we showed that a fair classifier constructed using Reweighing on the original data successfully reduced disparities between protected and privileged groups. Now, we investigate the scenario where the fairness correction occurs after the data has been privatized.

4.1 Implementing the Private+Fair Classifier

We start with the private dataset created in Chapter 3, where Age and Sex were perturbed using randomized response under a chosen ϵ (e.g., 0.5). On this privately perturbed dataset, we apply the Reweighing technique. The Reweighing algorithm attempts to rebalance the training data so that underprivileged groups are fairly represented, adjusting instance weights accordingly.

However, since the sensitive attributes are now noisy, the Reweighing method may not accurately identify which individuals belong to the underprivileged groups. This mismatch can reduce the effectiveness of the fairness correction. Despite this, we proceed to train the private+fair classifier on the reweighted private dataset and evaluate its fairness using the true sensitive attributes at test time (as if we are auditors with privileged access to the original data).

4.2 Results and Analysis

4.2.1 Classifier Performance

The private+fair classifier maintains a performance quite similar to that of the private classifier, as seen in its classification report. While Precision and F1-scores are only slightly affected, suggesting that adding fairness mitigation on top of privacy did not drastically degrade predictive performance, our primary interest lies in fairness metrics across different subgroups.

Class	Precision	Recall	F1-Score	Support
$\leq 50K$	0.89	0.94	0.91	4527
$> 50K$	0.78	0.64	0.70	1501
Accuracy		0.87		6028
Macro avg	0.84	0.79	0.81	6028
Weighted avg	0.86	0.87	0.86	6028

Table 4.1: Classification report of the private+fair classifier on the test dataset

4.2.2 Fairness Metrics on Real Sensitive Attributes

We measure fairness using the original (real) sensitive attributes in the test set. Specifically, we compute metrics like True Positive Rate (TPR), False Positive Rate (FPR), and Disparate Impact (percentage benefiting) for the same protected groups as before: Young vs. Senior, Female vs. Male, and the intersectional group Young Female vs. Senior Male.

	Accuracy	TPR	FPR	FNR	Disparate Impact
Senior	0.841781	0.641165	0.070966	0.358835	0.243792
Young	0.971429	0.541667	0.019452	0.458333	0.030303
Ratio	1.154015	0.844817	0.274101	1.277280	0.124299

Table 4.2: Fairness metrics by Age group for the private+fair classifier

Fairness Metrics by Age Category: For Age groups, the TPR ratio (Young/Senior) is about 0.84. While this shows some improvement over a purely private classifier scenario, it does not fully match the improvements we saw when fairness was applied without the privacy perturbation.

	Accuracy	TPR	FPR	FNR	Disparate Impact
Male	0.833128	0.645800	0.082471	0.354200	0.257445
Female	0.935878	0.606695	0.018540	0.393305	0.090076
Ratio	1.123330	0.939446	0.224807	1.110406	0.349885

Table 4.3: Fairness metrics by Sex group for the private+fair classifier

Fairness Metrics by Sex Category: For Sex, the TPR ratio (Female/Male) is approximately 0.94, which is relatively high but still not equalized. The Disparate Impact, at about 0.35, shows the female group still receives fewer positive outcomes proportionally compared to males.

	Accuracy	**TPR**	**FPR**	**FNR**	**Disparate Impact**
Senior Male	0.807986	0.647106	0.099445	0.352894	0.299472
Young Female	0.981928	0.500000	0.012195	0.500000	0.018072
Ratio	1.215278	0.772671	0.122632	1.416856	0.060347

Table 4.4: Fairness metrics for Senior Male vs. Young Female groups for the private+fair classifier

Fairness Metrics for Senior Male vs. Young Female: When examining these intersectional subgroups, the TPR ratio (Young Female / Senior Male) is about 0.77, indicating persistent disparities.

4.3 Results comparison

In the results of measuring the fairness of the fair+private model, we observed the deterioration in the true positive rate in comparison to the one obtained for the fair classifier. The results can be explained by the fact that the reweighing process is impacted by the prior application of the randomized response. The randomized response gives the false values for certain rows, which, in fact, does not correspond to the actual value. Thus, the reweighing process reweights these kind of rows falsely, which leads to deteoration in the fairness of the model.

5 Explainability

Task description:

> Study the explainability of the private classifier. Identify instances where the model is wrong but highly confident, and explain them. Assume you have access to the real sensitive values of Age and Sex. Investigate whether the noisy values for these attributes are responsible for the model being confident and wrong.

5.1 Preprocessing

Sensitive features (`age` and `sex`) were privatized using Local Differential Privacy (LDP) with $\epsilon = 0.5$ for `Private Classifier`.

To facilitate training and explainability:

- Data was preprocessed using the `Tabular` and `TabularTransform` tools from `omnixai`.
- Features were transformed for compatibility with the `Private Classifier`.
- Inverse transformations were applied to categorical or numerical formats during explanation.

5.2 Explainability Assessment

5.2.1 Global Explainability

Global explainability was performed using:

- **ALE (Accumulated Local Effects)**: Visualize how features influence the average prediction of a model.
- **SHAP (Shapley Additive Explanations)**: Measures the contribution of each feature to the model's predictions.

Visualizations are provided for ALE results in Figure 5.1.

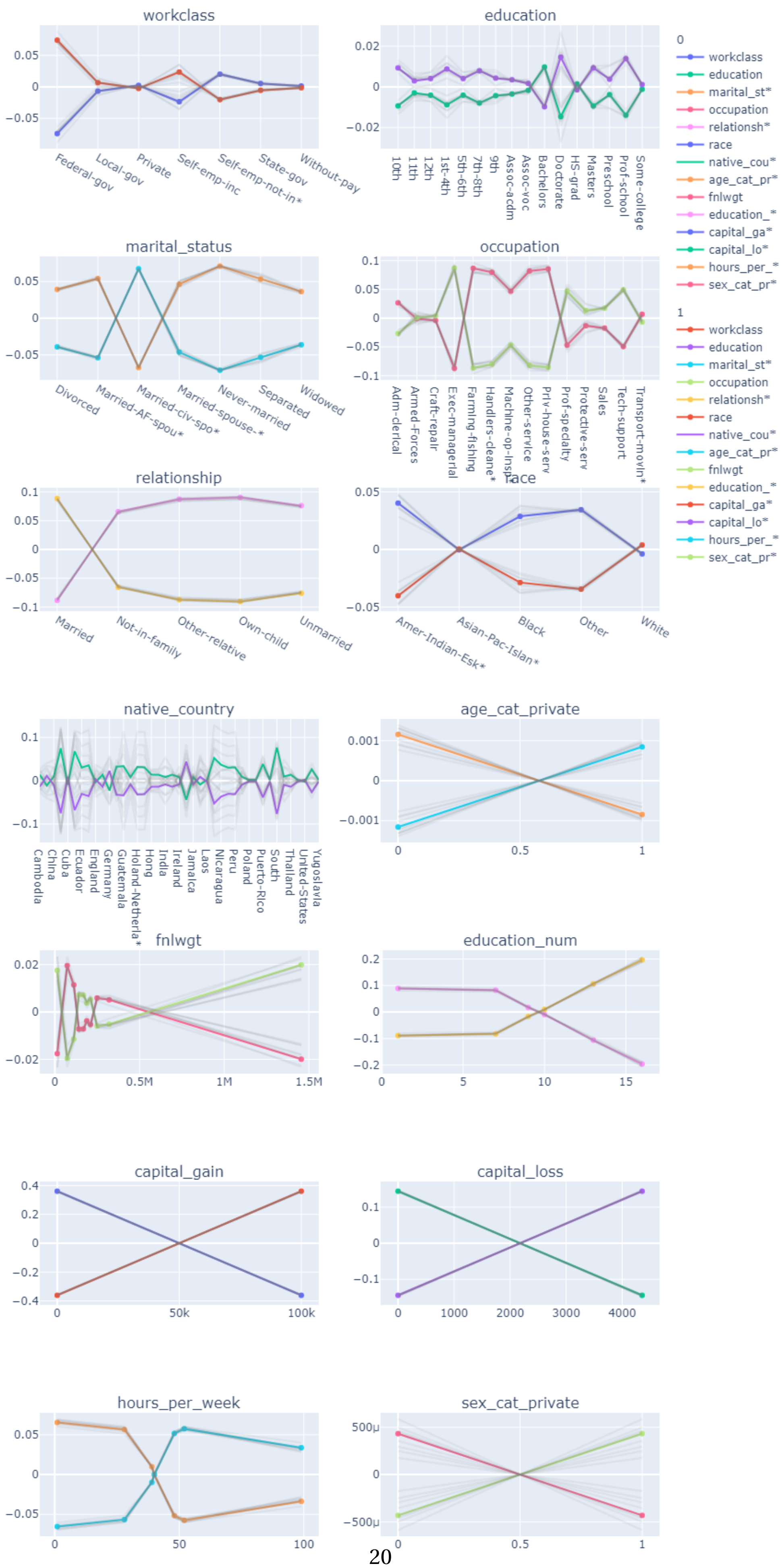

workclass
education
marital_status
occupation
relationship
race
native_country
age_cat_private
fnlwgt
education_num
capital_gain
capital_loss
hours_per_week
sex_cat_private
0
workclass
education
marital_st*
occupation
relationsh*
race
native_cou*
age_cat_pr*
fnlwgt
education_*
capital_ga*
capital_lo*
hours_per_*
sex_cat_pr*
1
Federal-gov
Local-gov
Private
Self-emp-inc
Self-emp-not-in*
State-gov
Without-pay
Divorced
Married-AF-spou*
Married-civ-spo*
Married-spouse-*
Never-married
Separated
Widowed
Married
Not-in-family
Other-relative
Own-child
Unmarried
Amer-Indian-Esk*
Asian-Pac-Islan*
Black
Other
White

Table 5.1 provides a detailed summary of the ALE (Accumulated Local Effects) results for each feature in the income prediction model. It highlights the impact of individual features on the model's predictions and offers an interpretation of these effects to understand their influence on predicting whether an individual's income exceeds $50K or not.

Table 5.1: Summary of ALE Results for Feature Impacts

Feature	**Impact on Income Prediction**	**Discussion**
Workclass	Federal-gov, Self-emp-inc positively impact income predictions, while Without-pay and Self-emp-not-inc negatively impact.	Reflects job stability and earning potential of different work environments.
Education	Higher education (e.g., Doctorate, Prof-school) has positive effects, however unexpected positive effects are observed for lower levels	Higher education levels are associated with access to better-paying jobs.
Marital Status	Married-civ-spouse strongly increases the likelihood of $> 50K$, while Never-married and Divorced reduce it.	Indicates financial stability in traditional marriage structures.
Occupation	Managerial and Professional roles positively affect income predictions $> 50K$, while lower-skilled jobs (like farming-fishing, handler-cleaner) negatively affect.	Demonstrates income disparity across job roles.
Relationship	Maried category positively impact income $> 50K$, while Not-in-family and Other-relative negatively impact.	Suggests the role of traditional family earners in income disparities.
Race	Some racial categories positively influence predictions, while others negatively influence.	Suggests potential biases or structural inequalities.
Native Country	The effect fluctuates.	Economic disparities based on country of origin.
Age (age_cat_private)	Very small y-axis values. This feature has minimal impact on the model's predictions.	Age is encoded as private makes the marginal effect negligible.
Fnlwgt	Effects are fluctuate based on weights.	Likely reflects sampling design effects in the dataset.

Education_num	Higher numerical education levels show positive effects on predictions.	Supports findings from the "Education" feature.
Capital Gain	High capital gain strongly positively influences predictions. Zero or low values negatively influence income predictions.	Indicates wealth or investment income as a strong income indicator.
Capital Loss	High values slightly positively impact income predictions, while zero has a neutral effect.	May reflect active investment activity typical of high-income earners.
Hours per Week	40-100 hours/week positively influence predictions, while fewer hours negatively impact.	Standard full-time work hours are associated with higher wages.
Sex (sex_cat_private)	Very small y-axis values. This feature has minimal impact on the model's predictions.	Sex is encoded as private makes the marginal effect negligible.

The SHAP global feature importance results show that the features workclass and education both have equal importance in determining the prediction for income >50K, with a SHAP score of approximately 0.236 for both, as shown in Figure 5.2.

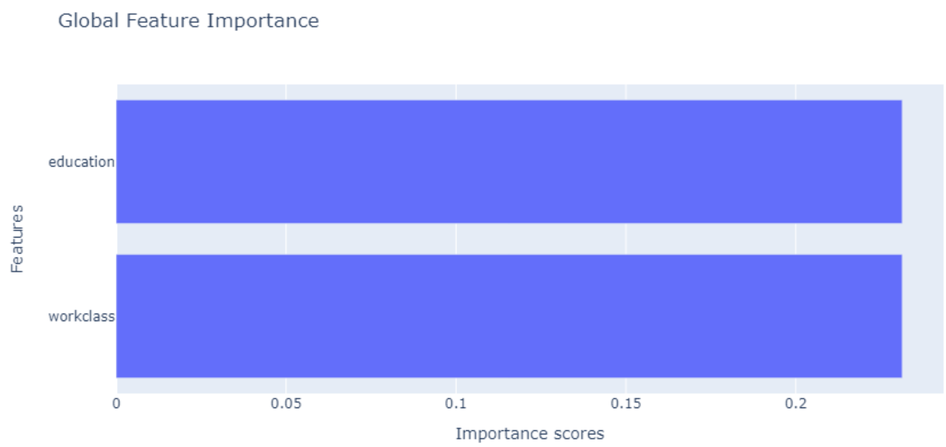


Figure 5.2: Global explainability results showing SHAP outputs.

5.2.2 Local Explainability

Localized insights were derived using:

- **LIME (Local Interpretable Model-agnostic Explanations)**: Highlights important features for individual predictions.

- **Counterfactual Explanations (MACE)**: Generates "what-if" scenarios to explore minimal changes required for a different prediction.

Local Explainability with LIME

We use the LIME (Local Interpretable Model-agnostic Explanations) method to understand how the model makes predictions for individual instances. We first randomly select five instances from the test set to generate local explanations for them. This randomness ensures that we explore a diverse set of data points and see how the model behaves across different scenarios.

For instance, consider Instance 27 (Figure 5.3, which has the true label 0 (i.e, income $\leq 50K$) and a predicted label of 0 (the model correctly predict). The prediction is accompanied by probabilities of 73.61% for $\leq 50K$ and 26.39% for >50K, indicating some level of uncertainty in the prediction.

Figure 5.3: Local explainability results showing LIME outputs for instance 27.

The LIME explainer provides insight into which features contributed to this prediction, with the following positive and negative contributions. For example: **Capital Gain** = 0 with score 0.72 strongly contribute to the prediction, high capital gains are typically associated with higher income, so having no capital gain lowers the likelihood of earning more than $50,000. Meanwhile the **hours per week** = 40 slightly contribute toward to the prediction. The features: **Marital Status** = *Married-Civ-Spo*, **relationship** = *married*, .etc, negatively contribute to the prediction.

Another example is Instance 3183 (Figure 5.4), which has the true label 1 (i.e, income >50K) and a predicted label of 1 (the model correctly predict). The prediction is accompanied by probabilities of 33.07% for $\leq 50K$ and 66.93% for >50K, indicating some level of uncertainty in the prediction.

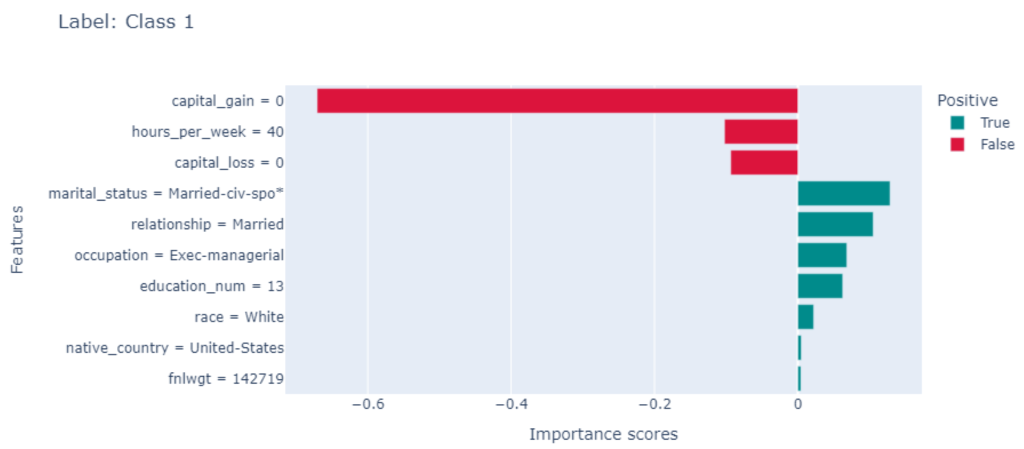


Figure 5.4: Local explainability results showing LIME outputs for instance 3183.

The fact that the individual has no capital gain (**capital gain** = 0) contributes negatively towards the >50K prediction and the individual working 40 hours per week contributes slightly to the false positive towards $\leq 50K$. Meanwhile, **Marital** and **Relationship** = *Married*, **Occupation** = *Exec-Managerial* contribute positively to the prediction.

Counterfactual

Counterfactual explanations explore how small changes in an individual's features can alter a model's prediction. Below, we analyze counterfactual results for two instances (e.g: Instances 27 & 3183) to illustrate how these adjustments influence the model's predictions.

Instance 27, in Figure 5.5 with the original prdiction that income $\leq 50K$:

- **Workclass**: Changed from *State-gov* to *Federal-gov*, resulting in a prediction of $> 50K$.
- **Capital_loss**: A significant increase from 0 to 1902 strongly influenced the prediction towards $> 50K$.
- Slightly change the **fnlwgt** and **education_num**, with the change in **occupation** from *Craft-repair* to *Exec-managerial* make the prediction $> 50K$.

The model is sensitive to government job roles, occupation and capital loss indicators, associating these with higher income potential.

workclass	fnlwgt	education	education_num	marital_status	occupation	relationship	race	capital_gain	capital_loss	hours_per_week	native_country	age_cat_private	sex_cat_private	label
State-gov	250976.000000	HS-grad	9.000000	Married-civ-spouse	Craft-repair	Married	White	0.000000	0.000000	40.000000	United-States	1	1.000000	0
Federal-gov	250976.000000	HS-grad	9.000000	Married-civ-spouse	Craft-repair	Married	White	0.000000	0.000000	40.000000	United-States	1	1.000000	1
Federal-gov	250976.000000	HS-grad	9.062500	Married-civ-spouse	Craft-repair	Married	White	0.000000	0.000000	40.000000	United-States	1	1.000000	1
Federal-gov	249805.625000	HS-grad	9.000000	Married-civ-spouse	Craft-repair	Married	White	0.000000	0.000000	40.000000	United-States	1	1.000000	1
State-gov	249805.625000	HS-grad	9.562500	Married-civ-spouse	Exec-managerial	Married	White	0.000000	0.000000	41.562500	United-States	0	1.000000	1
State-gov	250976.000000	HS-grad	9.000000	Married-civ-spouse	Craft-repair	Married	White	0.000000	1902.000000	40.000000	United-States	0	1.000000	1

Figure 5.5: Counterfactual results for instance 27.

As for instance 3183, in Figure 5.6, with the original prediction that Income $> 50K$:

- Changing **Marital_status** from *Married-civ-spouse* to *Never-married* shifted the prediction to $\leq 50K$.
- Minor adjustments to **fnlwgt** and **hours_per_week** (e.g., reducing hours slightly from 40 to 39.875) also led to a prediction of $\leq 50K$.
- Altering the **Occupation** to less senior roles (e.g., *Adm-clerical* or *Prof-specialty*) further reinforced the shift to $\leq 50K$.
- Introducing a higher **capital_gain** (e.g., 196.0625) in CF[4] with the change of occupation returned a prediction of $\leq 50K$, showcasing the complexity of the model's decision boundaries.

The model associates marital status (*Married-civ-spouse*) and a stable executive occupation (*Exec-managerial*) with higher income predictions. However, even with positive features like capital gain, changes to occupation can lead to a lower-income prediction.

workclass	fnlwgt	education	education_num	marital_status	occupation	relationship	race	capital_gain	capital_loss	hours_per_week	native_country	age_cat_private	sex_cat_private	label
Self-emp-inc	142719.000000	Bachelors	13.000000	Married-civ-spouse	Exec-managerial	Married	White	0.000000	0.000000	40.000000	United-States	0	1.000000	1
Self-emp-inc	142719.000000	Bachelors	13.000000	Never-married	Exec-managerial	Married	White	0.000000	0.000000	40.000000	United-States	0	1.000000	0
Self-emp-inc	143044.031250	Bachelors	13.000000	Never-married	Exec-managerial	Married	White	0.000000	0.000000	40.000000	United-States	0	1.000000	0
Self-emp-inc	142719.000000	Bachelors	13.000000	Never-married	Exec-managerial	Married	White	0.000000	0.000000	39.875000	United-States	0	1.000000	0
Self-emp-inc	138834.812500	Bachelors	13.000000	Married-civ-spouse	Adm-clerical	Own-child	White	0.000000	0.000000	40.000000	United-States	0	1.000000	0
Self-emp-inc	142719.000000	Bachelors	12.875000	Married-civ-spouse	Prof-specialty	Own-child	White	196.062500	0.000000	40.000000	United-States	1	1.000000	0

Figure 5.6: Counterfactual results for instance 3183.

5.3 Wrong but Highly Confident Predictions

5.3.1 Instances Detection

Instances where the classifier made incorrect predictions with confidence > 95% were identified (Figure 5.7). These cases indicate potential model overconfidence.

```
There are 41 instances where the model is very confident but wrong.
For example, instance 146 has label 1 and prediction 0, with probs [0.99796013 0.00203987]
```

Figure 5.7: Predict instances where the model is wrong but highly confident.

5.3.2 Explain

Feature contributions for these cases were examined using LIME and MACE. Below, we analyze two specific instances (146 and 1378) to understand the model's behavior and identify key factors leading to the high-confidence mispredictions.

Instance 146

LIME explanation: Instance 146 has label 1 but prediction 0, with probs [0.9522281 0.0477719]

- **True Label**: 1 (Income >50K)
- **Predicted Label**: 0 (Income ≤50K)
- **Prediction Confidence**: [0.952 ($\leq 50K$), 0.048 (>50K)]

In the Figure 5.8, LIME explains the model's decision as heavily influenced by *Capital Gain = 0*, which strongly favors the prediction of $\leq 50K$. Other factors (marital status, relationship) such as *Never-married* and *Not-in family* contribute to the misclassification, as these demographics are associated with lower incomes. *Hours per Week = 50* and *Occupation = Exec-managerial* push the prediction slightly toward $> 50K$, aligning with higher-income characteristics. However, these factors are insufficient to override the dominant influence of *Capital Gain = 0*.

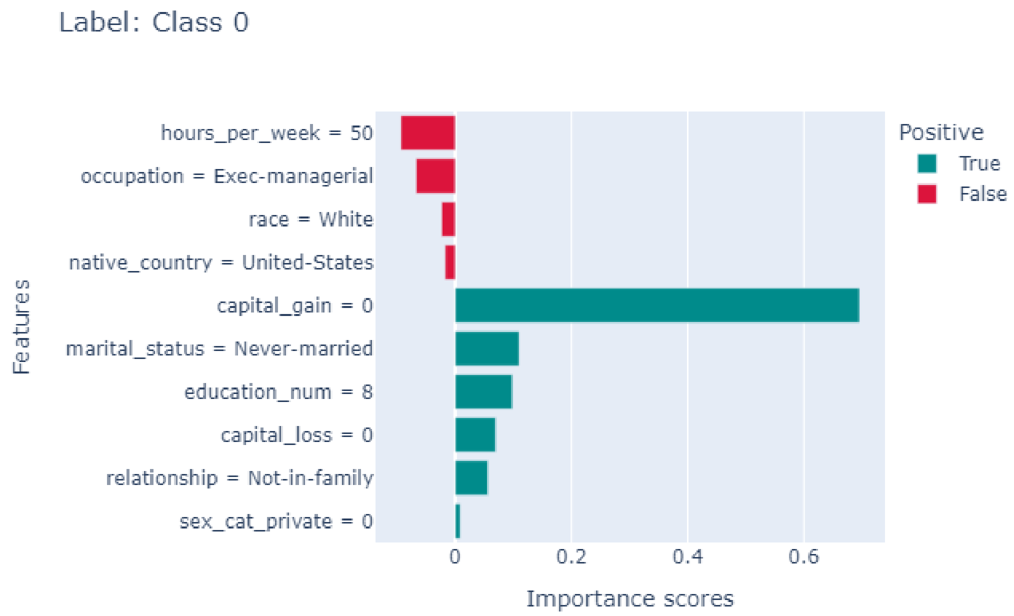


Figure 5.8: LIME explanation of Instance 146.

In Figure 5.9, MACE counterfactual suggest specific changes to this instance to achieve a prediction of $> 50K$:

- Change *Marital Status* from *Never-married* to *Married-civ-spouse*.
- Slightly increase *Education Number* to 8.0625.
- Change *Workclass* to *Self-employed* category.
- Increase *Capital Loss* values (e.g., 2399 or 2559).

workclass	fnlwgt	education	education_num	marital_status	occupation	relationship	race	capital_gain	capital_loss	hours_per_week	native_country	age_cat_private	sex_cat_private	label
State-gov	186634.000000	12th	8.000000	Never-married	Exec-managerial	Not-in-family	White	0.000000	0.000000	50.000000	United-States	1	0.000000	0
State-gov	186634.000000	12th	8.000000	Married-civ-spouse	Exec-managerial	Married	White	0.000000	0.000000	50.000000	United-States	1	0.000000	1
State-gov	186634.000000	12th	8.062500	Married-civ-spouse	Exec-managerial	Married	White	0.000000	0.000000	50.000000	United-States	1	0.000000	1
Self-emp-inc	186634.000000	12th	8.000000	Married-civ-spouse	Exec-managerial	Married	White	0.000000	0.000000	50.000000	United-States	1	0.000000	1
Self-emp-not-inc	186634.000000	12th	8.000000	Never-married	Exec-managerial	Not-in-family	White	0.000000	2399.062500	50.000000	United-States	1	0.000000	1
State-gov	186634.000000	HS-grad	7.875000	Never-married	Exec-managerial	Not-in-family	White	0.000000	2559.000000	49.375000	United-States	0	0.000000	1

Figure 5.9: MACE explanation of Instance 146.

The model is highly influenced by the absence of *Capital Loss* as well as the marital status of *Never-married*, leading to a misclassification. Counterfactual analysis shows that changing marital status or adding capital loss shifts the prediction significantly.

Instance 1378

LIME explanation: Instance 1378 has label 0 and prediction 1, with probs [0.00320717 0.99679283]

- **True Label**: 0 (Income ≤50K)
- **Predicted Label**: 1 (Income >50K)
- **Prediction Confidence**: $[0.997\ (>50K), 0.003\ (\leq 50K)]$

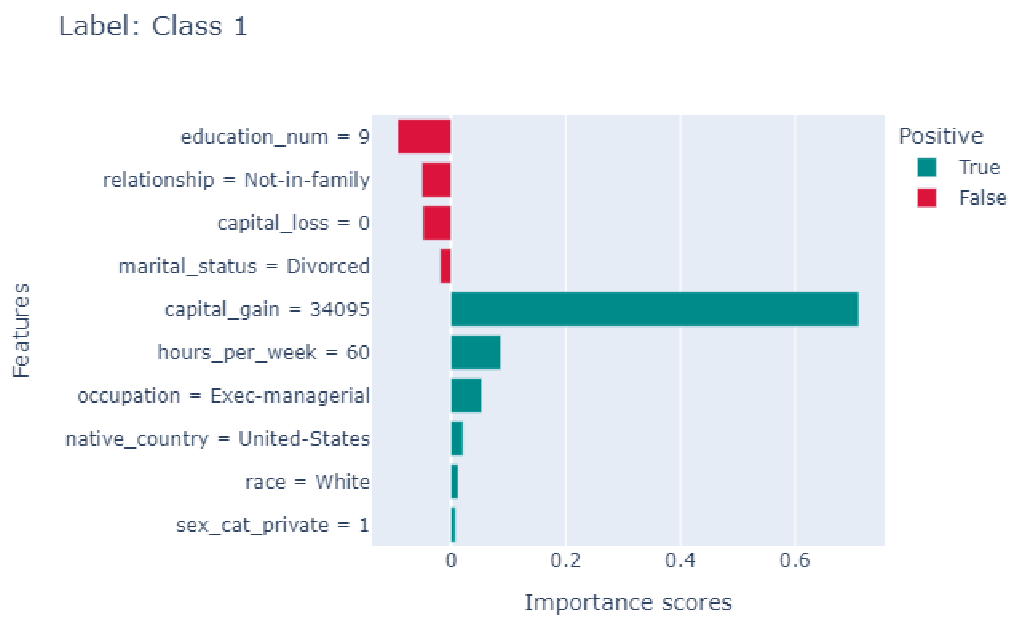


Figure 5.10: LIME explanation of Instance 1378.

The most influential feature is *Capital Gain = 34095*, which heavily biases the prediction towards high income, with a score of 0.713. Other supporting features include *Hours per Week = 60* and *Occupation =*

Exec-managerial, typically associated with high earners. Meanwhile, features such as *Education Num = 9*, *Relationship = Not-in-family*, and *Marital Status = Divorced* negatively impact the prediction, reflecting characteristics more typical of lower-income groups. However, their combined influence is outweighed by the dominant positive contributors.

In Figure 5.11, MACE counterfactual suggest specific changes to this instance to achieve a prediction of $\leq 50K$: A substantial reduction in only *Capital Gain* feature is sufficient to correct the prediction, even while maintaining other features that suggest higher income. This indicates that the model places excessive importance on capital-gain, potentially leading to misclassifications for individuals with atypical financial situations.

workclass	fnlwgt	education	education_num	marital_status	occupation	relationship	race	capital_gain	capital_loss	hours_per_week	native_country	age_cat_private	sex_cat_private	label
Self-emp-not-inc	145574.000000	HS-grad	9.000000	Divorced	Exec-managerial	Not-in-family	White	34095.000000	0.000000	60.000000	United-States	1	1.000000	1
Self-emp-not-inc	145574.000000	HS-grad	9.000000	Divorced	Exec-managerial	Not-in-family	White	6392.812500	0.000000	60.000000	United-States	1	1.000000	0

Figure 5.11: MACE explanation of Instance 1378.

5.4 Impact of Noisy Sensitive Attributes on Model Confidence and Errors

To investigate the contribution of noisy privatized values to the model's wrong predictions, we test the current model (trained on privatized data) on both the privatized dataset and the real dataset. We have considered a second approach, which is to train two models on the privatized dataset and the real dataset, respectively, and investigate their performances. However, we limit the scope of this project to the first approach with the assumption that we are investigating the performances of a trained model, and that the real sensitive values of Age and Sex are only accessible during the validation phase.

5.4.1 Impact on wrong and highly confident data instances

First, we focus on the same data instances identified in 5.3.1 to investigate whether the noisy values have an impact on data instances where the model is wrong and highly confident. Out of all 41 instances where the model is wrong and highly confident, a few of them have different Age and Sex values between the privatized data and the real data, but notably, none of them get labeled differently when testing on the two sets. Figure 5.12 shows a portion of these data instances and visualizes this observation more clearly. **So for these data instances, the noisy values of Age and Sex do not seem to impact the model's predictions.**

Instance ID	Preds on private set	Probs on private set	Preds on real set	Probs on real set	Actual labels	Private values	Real values	delta
146	0	0.950974	0	0.950974	1	age=0, sex=0	age=0, sex=0	0.000000
257	0	0.962612	0	0.962612	1	age=1, sex=0	age=1, sex=0	0.000000
300	0	0.980998	0	0.985368	1	age=0, sex=0	age=1, sex=0	0.004369
638	0	0.987381	0	0.987074	1	age=0, sex=0	age=1, sex=0	-0.000307
1040	0	0.990161	0	0.990161	1	age=1, sex=1	age=1, sex=0	0.000000
1103	0	0.953777	0	0.953777	1	age=1, sex=1	age=1, sex=0	0.000000
1317	0	0.960916	0	0.950598	1	age=0, sex=0	age=1, sex=0	-0.010318
1378	1	0.995470	1	0.995470	0	age=1, sex=0	age=1, sex=0	0.000000
1409	0	0.965578	0	0.965578	1	age=1, sex=0	age=1, sex=0	0.000000
1426	0	0.961467	0	0.958530	1	age=0, sex=1	age=1, sex=0	-0.002936
1481	0	0.957093	0	0.956354	1	age=1, sex=1	age=1, sex=0	-0.000740
1528	0	0.984361	0	0.982529	1	age=0, sex=0	age=1, sex=0	-0.001832
1640	0	0.986277	0	0.986277	1	age=1, sex=1	age=1, sex=0	0.000000
1847	0	0.979601	0	0.979601	1	age=1, sex=1	age=1, sex=0	0.000000
2142	0	0.965419	0	0.965419	1	age=1, sex=0	age=1, sex=0	0.000000
2344	0	0.957986	0	0.957986	1	age=1, sex=0	age=1, sex=0	0.000000
2351	0	0.972546	0	0.972546	1	age=1, sex=0	age=1, sex=0	0.000000
2407	0	0.978135	0	0.978135	1	age=1, sex=0	age=1, sex=0	0.000000
2530	0	0.993354	0	0.992653	1	age=0, sex=1	age=1, sex=0	-0.000701
2671	0	0.956057	0	0.954668	1	age=1, sex=0	age=1, sex=0	-0.001390
2742	1	0.962647	1	0.962647	0	age=1, sex=0	age=1, sex=0	0.000000

Figure 5.12: Model prediction comparison between privatized and real data for wrong and highly confident data instances.

Now, to investigate whether this has an impact on the model's confidence, we also calculate the delta values, defined as the prediction probability on the real data minus the prediction probability on the privatized data. Table 5.2 summarizes some statistics for the delta column. It is observable from the statistics as well as from Figure 5.12 that the delta values are relatively small across the instances. In other words, **the model's confidence for these data instances does not vary significantly between the privatized and real datasets**. For some instances, there is virtually no change in the prediction probability, even with the noisy Age and Sex values. An example of this is the instance with ID 1040, where the sex value (0) was privatized as (1) but the prediction probability remains unchanged.

Mean	-0.000625
Standard deviation	0.002469
Min	-0.010637
25%	-0.000711
50%	0.000000
75%	0.000000
Max	0.004369

Table 5.2: Statistics of the delta column

5.4.2 Instances with label change

Next, we look into the data instances whose labels do vary between testing on the privatized data and testing on the real data. There are 25 such instances. Figure 5.13 shows a sample of them, where for each pair of rows with the same instance ID (first column), the first row shows the privatized Age and Sex values while the second row shows the real values. The prediction probabilities for the two labels are also shown, as well as the true label.

	age_cat_private	sex_cat_private	pred	prob_0	prob_1	true_label
739	0	1.0	0	0.504902	0.495098	1
739	1	0.0	1	0.425846	0.574154	1
888	0	0.0	0	0.519332	0.480668	0
888	1	0.0	1	0.492794	0.507206	0
1244	0	0.0	1	0.495741	0.504259	0
1244	1	0.0	0	0.501124	0.498876	0
1325	0	0.0	1	0.472264	0.527736	0
1325	1	0.0	0	0.578866	0.421134	0
1560	0	0.0	0	0.503866	0.496134	1
1560	1	0.0	1	0.487658	0.512342	1
1761	0	1.0	0	0.501826	0.498174	0
1761	1	0.0	1	0.477149	0.522851	0
2193	0	1.0	1	0.477603	0.522397	0
2193	1	0.0	0	0.503967	0.496033	0
2490	0	1.0	1	0.446611	0.553389	0
2490	1	0.0	0	0.515631	0.484369	0
2564	0	0.0	0	0.501950	0.498050	1
2564	1	0.0	1	0.410039	0.589961	1
2598	0	1.0	0	0.520782	0.479218	1
2598	1	0.0	1	0.463339	0.536661	1
2677	1	1.0	1	0.488487	0.511513	0
2677	1	0.0	0	0.523840	0.476160	0

Figure 5.13: Model prediction comparison between privatized and real data for instances where prediction changes.

It is easily observable that the model is not confident when predicting these data instances, whether with the privatized or the real data, which is evident in the fact that the prediction probabilities are all very close to 0.5 for both labels. The noisy values of Age and Sex have an impact on these predictions by changing the probabilities by a small amount, which in this case is barely enough to flip the model's decisions. In other words, the noisy values of the sensitive attributes are only influential in cases where the model is already not confident.

6 Explainability and LLMs

Task description:

> Select an explainability method and create a natural language interface to it. The idea is to take an explanation (e.g., feature-importance pairs, examples) and present it as text that a person can easily understand.

6.1 Explainability Method

For this project, we focus on the feature importance explanation method, which provides insights into how a machine learning classification model makes predictions. Our chosen approach involves:

- Extracting feature importance scores
- Formatting a prompt
- Request human-readable interpretations from LLM

6.2 Implementation

6.2.1 Extracting feature importance scores

First, we extract the local explanations using LIME (Local Interpretable Model-agnostic Explanations). The following code retrieves the first explanation from the local explanations:

```
local_explanations = explainers.explain(X=...)
ins = local_explanations['lime'].get_explanations()[0]
```

The explanation has the following components:

- instance: the specific data point being explained
- features: input features contributing to the prediction
- values: feature values

- scores: importance scores for each feature
- target_label: the predicted class or label

Extracting the values from the information, we format the feature - value - score tuple as `"feature = value": score`, for example:

```
0. "capital_gain = 0.0": score=0.7409
1. "marital_status = Never-married": score=0.1091
2. "hours_per_week = 50.0": score=-0.0869
3. "education_num = 8.0": score=0.0825
4. "capital_loss = 0.0": score=0.0668
5. "occupation = Exec-managerial": score=-0.0570
6. "relationship = Not-in-family": score=0.0412
7. "education = 12th": score=0.0349
8. "workclass = State-gov": score=0.0179
9. "native_country = United-States": score=-0.0172
```

6.2.2 Formatting a prompt

In this subsection, we incorporate the previously extracted features and importance scores into a prompt designed to guide the language model in generating interpretable explanations of the machine learning model's predictions.

Below is an example of a prompt sent to the LLM to request explanations for a specific data point:

```
You are a helpful assistant for explaining prediction results generated by a machine
    learning classification model based on the information provided below. Your
    answers should be accurate and concise. Firstly, given the following feature
    importance scores in the format ""feature=value": feature importance score":

"capital_gain = 0.0": score=0.7197
"marital_status = Never-married": score=0.1095
"education_num = 8.0": score=0.0930
"hours_per_week = 50.0": score=-0.0708
"capital_loss = 0.0": score=0.0605
"relationship = Not-in-family": score=0.0551
"occupation = Exec-managerial": score=-0.0537
"workclass = State-gov": score=0.0279
"education = 12th": score=0.0232
"race = White": score=-0.0178

Please explain how these features impact the classifier's decision to label this
    example as label_0. Also pay attention to the values associated with the features
    .
```

Listing 6.1: Example of a prompt

6.2.3 Request human-readable interpretations from LLM

We then set up a connection to **Llama 3.2 1B** using LLM Studio for generating explanatory text:

```python
from openai import OpenAI

client = OpenAI(
    api_key="your-api-key", # not needed
    base_url="http://localhost:1234/v1"
)
messages = [
    {"role": "system", "content": "You are a helpful ML explanation assistant!"}
]

def get_response(message, messages):
    messages.append({"role": "user", "content": message})
    completion = client.chat.completions.create(
        messages=messages,
        temperature=0.8,
        model="model"
    )
    response = completion.choices[0].message.content
    messages.append({"role": "assistant", "content": response})
    return response, messages

message =  context_prompt
response, messages = get_response(message, messages)
```

Listing 6.2: Establishing connection with LLM via LM Studio

6.3 LLM responses

Using the prompts crafted in the manner described above, we receive comprehensible interpretations. One such example is shown below:

> To predict a label of label_0 based on the given feature importance scores, we need to consider the following:
>
> - Capital Gain: This feature has the highest score (score = 0.7197), indicating it's considered most important in determining whether to label as label_0. However, having a high score doesn't guarantee labeling, but it suggests that capital gain is a significant factor.
> - Marital Status: The second highest importance score (score = 0.1095) means marital status also

plays a role, although less so than capital gain. Being "Never-married" might increase the likelihood of label_0.

- Education Number: An education level of 8 has an importance score close to 0.0930, suggesting that educational background is slightly less important in this context.
- Hours Per Week: Negative value (score = -0.0708) indicates hours per week are not considered a significant factor for labeling label_0.
- Capital Loss: A very low score (-0.0605) implies capital loss is not a deciding factor for the classifier to label as label_0.
- Relationship: The feature with an importance score of 0.0551 is "Not-in-family", which is also considered less important than marital status and capital gain in this case.
- Occupation: An occupation level (score = -0.0537) has a relatively low score, suggesting it's not a critical factor for labeling label_0.
- Workclass: Being classified as "State-gov" has an importance score of 0.0279, which is lower than the other features.
- Education Level: Having an education level of 12th (score = 0.0232) indicates a certain educational background, but it's not considered significant in determining whether to label as label_0.
- Race: A negative score (-0.0178) for "White" suggests that racial background may be a factor, although the impact is relatively small compared to other features.

Considering these scores and values:

- Capital gain (score = 0.7197) seems most important.
- Marital status (score = 0.1095) is also significant but less so than capital gain.
- Educational background is slightly less important.
- Racial background plays a minor role.

Given these considerations, the classifier's decision to label as label_0 would likely be influenced by a mix of factors, with capital gain being the most decisive feature.

6.3.1 Discussions

Overall, the LLMs available for use via LM Studio are generally sufficient for light-weight workloads such as the interpretation task described in this chapter. However, their responses are of course far from perfect. The experiments we conducted show that the models tend to extrapolate and create new information without explicit contextual data provided in the prompt. In use cases such as this exercise where we task the LLM with interpreting our provided inputs, this proves to be a challenge as we face some issues with constraining the response within the scope of the explanations. The LLMs also seem to struggle with comprehending quantities and other numerical and mathematical values, as well as with drawing connections between the numerical values and their significance with regards to the feature importance. Therefore, even if our experiments show some promising results with using LLMs to generate comprehensible explanations, we would recommend sufficient skepticism in accepting its responses. A number of other examples on our experiments with LLMs and their pitfalls are discussed in the next chapter.

7 Other insights

7.1 LLM as a tool for generating easily comprehensible explanations

In the previous chapter, we explore the usage of LLMs in creating a human-comprehensible interpretation of model explanations. In addition to using LIME as previously shown, we also experimented with generating interpretations of Counterfactual Explanations (MACE) explanations. Below is an example of a prompt that we used:

```
You are a helpful assistant for explaining machine learning classification model predictions using Counterfactual Explanations (MACE). The classifier predicts whether someone will have income >50K (label_1) or not (label_0). Your task is to provide a clear and concise interpretation of the counterfactual explanations.

Original Instance (ID: 5704):
- workclass: Self-emp-not-inc
- fnlwgt: 388496.0
- education: Some-college
- education_num: 10.0
- marital_status: Divorced
- occupation: Craft-repair
- relationship: Not-in-family
- race: White
- capital_gain: 0.0
- capital_loss: 0.0
- hours_per_week: 8.0
- native_country: Puerto-Rico
- age_cat_private: 1
- sex_cat_private: 0.0
- label: 0

Counterfactual Explanations:

Counterfactual 1:
```

```
- capital_gain: 7290.9375 (changed from 0.0)
- label: 1 (changed from 0)

Please explain:
1. The key features that contribute to the original prediction
2. How the changes in the counterfactual instances might alter the prediction
3. Insights into the model's decision boundary based on these counterfactuals
```

We observe that the LLM tends to interpret the features wrongly, leading to very confident but completely unsupported responses such as:

> Based on the original instance ID 5704, the following features contribute to the predicted label of 0:
>
> 1. workclass: Self-emp-not-inc (a self-employed individual without corporate income)
> 2. fnlwgt: 388496.0 (a relatively high income level)
> 3. education: Some-college (some college degree, indicating a possible career advancement opportunity)
> 4. age_cat_private: 1 (private sector workers aged between 20 and 39 years old)

The following points are wrong and are not based on any provided context:

- fnlwgt as an indicator of income level
- age_cat_private as an indicator of sectors

Attempting to reduce this, we added extra contextual information, especially extra information on what fnlwgt and age_cat_private represent. Using external references[1], we add the following description to the prompt:

```
This is what the features mean:
    age_cat_private: The age of an individual, categorized as over 25 (1) or not (0)
    workclass: A general term representing the employment status of an individual
    fnlwgt: The final weight representing the number of people the census believes
   this entry represents.
    education: The highest level of education achieved by an individual
    education_num: A numerical representation of the highest level of education
   achieved
    marital_status: The marital status of an individual, including specific
   categories
    occupation: A general type of occupation for an individual
    relationship: Represents what this individual is relative to others
    race: Descriptions of an individual's race
    sex_cat_private: The sex of the individual, categorized as male (1) or female (0)
```

[1] https://cseweb.ucsd.edu/classes/sp15/cse190-c/reports/sp15/048.pdf

```
    capital_gain: The capital gains reported for an individual, represents the amount of money an individual has gained from the sale of investments such as stocks, bonds, or real estate.
    capital_loss: The capital loss reported for an individual, represents the amount of money an individual has lost from the sale of investments such as stocks, bonds, or real estate.
    hours_per_week: The number of hours an individual reports working per week.
```

Listing 7.1: Contextual information

Even then, the response tends to include hallucinations that are not supposed by any provided information in the prompt, for example:

> ...
>
> 1. "age_cat_private: 1": This indicates that the individual is private (score of -0.8) which suggests they might have lower income expectations.
>
> 2. "sex_cat_private: 0.0": This implies that the sex of the individual is private (score of -0.4), indicating a potential link to lower income expectations for this demographic group.
>
> ...

Additionally, it seems that the longer the prompt, the more the LLM tend to autonomously (and incorrect) extrapolate more information as to what the features indicate.

Since we were not able to reduce this behavior with MACE explanations, we decided not to include the results and opt for feature-importance pairs instead. This observation raises several questions regarding the well-known issue of hallucinating in LLMs and their limited ability in interpreting mathematical contents.

- To which extent should we entrust LLMs with tasks such as creating readable contents from highly technical ones?
- What are the measures we should put in place to ensure the model is being factual and accurate?
- How do we effectively establish the scope of context/knowledge that the model should comply to?
- Is prompt engineering sufficient for keeping the model factual and accurate?