Homework Understanding :

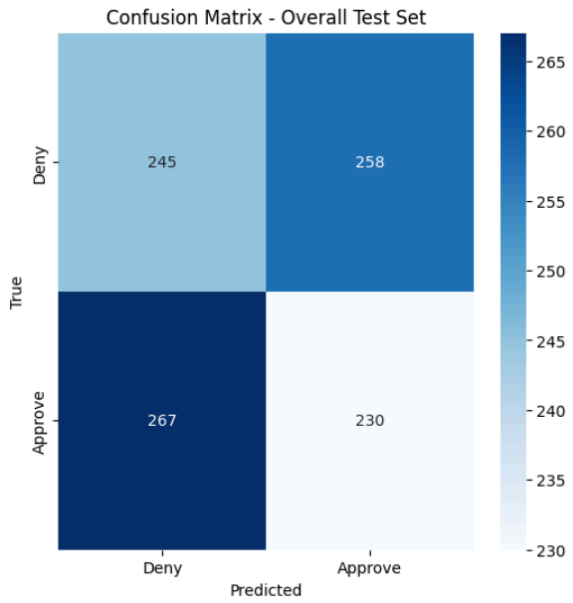
The task is to detect and correct bias in a poorly performing loan approval model on a new test set. We need to analyze potential biases in the model's predictions, depict them using confusion matrices, and suggest solutions to obtain more balanced results across different groups in the dataset.

Case Study 1

My approach:

I began by loading the pre-trained loan approval model from the provided .pkl file. The model, trained to predict loan approval based on various applicant features, was then tested on a new dataset (test_bias_new.csv). Upon running the model, I found the performance on the new data to be surprisingly low, with a test accuracy of only 47.5%. This poor performance raised concerns, suggesting that the model might be biased.

Given this low accuracy, I suspected that the model might be treating different groups unfairly, especially considering the potential for bias in decision-making processes like loan approval. Bias in machine learning models can lead to unequal treatment of certain groups, which is problematic in critical applications like this of loan giving. The confusion matrix for the overall test set shows that the model has a significant number of false positives (258) and false negatives (267), indicating poor decision-making. This suggests that the model is not making accurate loan approval decisions, which warrants further investigation into potential bias.



Steps that didnt help me:

1. Initially I tried to figure out the issue in the dataset by manually looking into the dataset for initial understanding but it didnt lead to any core analysis or didnt help me gather any thoughts.
2. I also tried analyzing the model's performance based on loan amount (a feature unrelated to gender or credit score). After subsetting the data based on high vs. low loan amounts, the confusion matrix showed no significant bias in performance, suggesting that loan amount did not introduce any bias. This confirmed that not every feature inherently introduces bias into the model's predictions.

Feature 1: Credit Score analyze_credit_score_bias(df, clf)

Steps taken to find the bias:

If we're looking for the most directly relevant feature for loan approval, credit score is the most important one to analyze for bias, as it directly impacts the approval decision.

I divided the dataset into two groups based on credit score: one for high credit score applicants (above 650) and one for low credit score applicants (650 or below).We selected 650 as the threshold for splitting applicants into high and low credit score groups because it is a common cutoff used in the financial industry. Typically, a credit score above 650 is considered good or low-risk, while scores below this threshold are often seen as high-risk. This value is widely used in models to distinguish between applicants who are more likely to repay loans and those who may pose a higher risk. Thus , for manual and initial understanding we do single querying and then confusion matrix evaluation to detect bias.

For initial querying we linearly queried the model for the first 10 applicants in both the high credit score and low credit score groups.The predictions were :

Predictions for High Credit Score Applicants (First 10): [0, 0, 0, 0, 0, 0, 0, 1, 1, 0]

Predictions for Low Credit Score Applicants (First 10): [1, 1, 1, 1, 1, 0, 0, 0, 0, 0]

The predictions for the high credit score applicants were a mix of approvals (1) and denials (0),in contrast to the predictions for the low credit score applicants were predominantly approvals (1).Hence , ths showed that model was more lenient towards low credit score applicants and more restrictive with high credit score applicants which might be a potential bias. Hence we looked into plotting the confusion metrics for credit score .

The output for that is :

Confusion Matrix for High Credit Score Applicants:

[[20 21]

 [266 230]]

Confusion Matrix for Low Credit Score Applicants:

[[225 237]

 [1   0]]

The confusion matrix for high credit score applicants shows that the model performs well, but it still denies many qualified applicants (266 false negatives) while making only a few false positives (21). This suggests that the model is overly cautious with high credit score applicants.

For low credit score applicants, the model is highly biased, denying almost all applicants (with only 1 false positive) and not approving any (0 true positives). This indicates a severe bias against low credit score applicants, as the model unfairly rejects them despite some potentially being eligible.

The model exhibits bias, favoring high credit score applicants and unfairly denying loans to low credit score applicants, regardless of their true eligibility.

How is it affecting ?

The bias of the model is affecting its performance by leading to biased decision-making based on credit score. For high-credit-score applicants, the model is too risk-averse, rejecting too many who would be approved, thus slowing down its ability to make right decisions for this group. For low-credit-score applicants, the model is extremely biased, rejecting almost all applicants and approving none, even though some are worthy of loans. This bias is affecting the model on a grand scale, as it leads to discriminatory treatment of low credit score applicants on a mass scale and over-approving of high credit score applicants. The overall correctness and fairness of the model is affected because of this imbalance.

Our Inferences

We identified bias in the model through a combination of confusion matrix analysis and single query testing. The confusion matrices revealed that the model was overly cautious with high credit score applicants, denying many who should have been approved, while showing a strong bias against low credit score applicants by denying almost all of them. Additionally, single query testing confirmed this bias, with the model approving most high credit score applicants and predominantly denying low credit

score applicants. Together, these analyses demonstrated that the model was unfairly favoring high credit score applicants and discriminating against low credit score applicants.

What is the extent of bias?

The bias of the model is good off. For customers with good credit score, the model is overly cautious, turning down 266 such customers who should have been accepted, a grave error in how it handles this segment. For low-credit-score applicants, the bias is more pronounced because the model accepts almost none of them, lending to only 1 applicant among 10 applicants, although they may be qualified. This portrays extreme bias toward low-credit-score applicants and a significant discrepancy in loan offering between the two groups. The model's performance is thus biased, and it makes discriminatory outputs, negating its fairness.Thuus , we check the other features to check bias .

Feature 2: Sex analyze_gender_bias(df, clf)

What are the steps taken to find bias ?

In order to check the likelihood of gender (sex) bias, I first separated the dataset into male and female applicants based on the sex column (0 for male, 1 for female). Then, I normalized features for both groups to the same format as the model's training data. I initially used single querying to use the model on the first 10 males and 10 females to observe the model's individual predictions. I subsequently used the confusion matrix method to observe if the model accurately predicted approvals or denials for males and females, allowing me to clearly ascertain if there was any bias in the model's decision-making.

What Was the Bias? How Was It Affecting the Model?

The bias was gender based , and the model exhibited a **strong tendency** to approve all male applicants while denying all female applicants. This resulted in unfair treatment between genders, with males being consistently approved for loans regardless of their eligibility, while females were entirely rejected by the model.

The single querying method just for testing gave us very surprising results , and hence we ended up seeing bias straight up , but since we tried for only 10 values (value 10 selected randomly) we need stronger proofs to prove our bias .This leads us to qworking on the confusion matrix.

The output for single querying is :

Predictions for Male Applicants (First 10):

[1, 1, 1, 1, 1, 1, 1, 1, 1, 1]

Predictions for Female Applicants (First 10):

[0, 0, 0, 0, 0, 0, 0, 0, 0, 0]

From the result we can see that the model is consistently approving all male applicants and denying all female applicants, which strongly suggests gender bias in the model. Thus we move to confusion matrix to check entire test set and give stronger proof to the boss.

BIAS ANALYSIS USING CONFUSION MATRIX

The confusion matrix obtained was as follows:

Confusion Matrix for Male Applicants:

[[0 258]

[0 230]]

Confusion Matrix for Female Applicants:

[[245 0]

[267 0]]

The bias was aptly demonstrated by male and female candidate confusion matrices. For male candidates, the confusion matrix indicated 0 false negatives (no males were rejected in error), but 258 false positives (too many males were accepted who should have been rejected). For women applicants, the matrix showed 0 true positives (no women were accepted), and 267 false negatives (all women applicants who would have been accepted were rejected in error). The absence of true positives for women and false positives for men further cemented the strong gender bias in the model. The heatmaps of these confusion matrices well illustrated this imbalance and showed the biased treatment of female applicants..

Comparision between Biased based on creditscore vs sex?

The gender bias is more severe than the credit score bias. For women applicants, the model refuses loans to all since there are 0 true positives and 267 false negatives. The credit score bias is less extreme, with the model being easy on low credit score applicants but still rejecting some high credit score applicants. The gender bias is a more discriminatory one, excluding all female applicants irrespective of their qualifications, and thus it is a more powerful form of bias than the credit score-related one.Hence, this leads to lower testing accuracy of the new dataset being tested.

CASE STUDY 2

My understanding :

In Case Study 2, the issue was to check and minimize possible bias in a logistic regression model to forecast whether a student is going to pass or fail a course based on three attributes: average homework grade, exam 1 score, and attendance percent. The model first performed on 69% train accuracy and 71% test accuracy, which reflected the possibility of bias in the data set.

Thus , I started reflecting on it to find bias initially , the simplest approach I could think of was to plot a confusion matrix and then carry on further analysis.

Also we need to provide steps for mitigating our detected bias .

Steps taken to find bias :

1.Checking class imbalance

When the distribution of classes is not balanced and one class has significantly more samples than the other the model can become biased toward predicting the majority class and it may come as an issue as the model may learn to always predict the majority class and this might come as an issue , thus we check it in starting itself .

The results were :

Training Data Class Distribution:

"Pass" (will_pass = 1)	1098 students
"Fail" (will_pass = 0)	502 students

Test Data Class Distribution:

"Pass" (will_pass = 1)	282 students

"Fail" (will_pass = 0)	118 students

The training set has a 2:1 pass to fail ratio (1098 vs. 502), reflecting class imbalance. The test set has a 282 vs. 118 ratio, reflecting a less severe but still existing imbalance.

The model, being trained on a high percentage of pass dataset, tends to predict "pass" more as it has come across more "pass" instances. This causes the model to bias towards the majority class and not correctly predict "fail" students.

The class imbalance likely explains the 71% test accuracy because the model may achieve it by simply marking pass students. However, it likely does not mark "fail" students, which compromises its overall performance.

We then move on to plot the confusion matrix .If the model is biased due to class imbalance, we could expect to see many false positives and possibly false negatives after which we can decide our next steps.

The output of the confusion matrix was :

[[0 118]

[0 282]]

We can see that it indicates that the model is biased towards predicting "pass" for every student, 118 false positives (failing students predicted to pass) and no false negatives or true negatives. This indicates a serious class imbalance issue, where the model cannot correctly predict "fail" students and hence it definitely hints towards class bias .

We then do the coefficient analysis so that we understand if a single feature is influencing more than the other and if yes then we can analyse the bias. The results of that were :

	Feature	Coefficient
0	average_HW_score	5.114427
1	exam1_score	5.446660
2	attendance_percentage	4.499181

Coefficient analysis alone doesn't reveal any obvious bias here because all attributes are accompanied by positive coefficients, indicating positive association with passing probability. That would mean the model is behaving consistently with all attributes assisting the prediction positively. All having positive coefficients and no observable divergences, it does not indicate towards any particular bias. But the class imbalance of the dataset is likely the key determinant of the performance of the model, with it inclined towards predicting the majority class ("pass") over the minority class ("fail").

Moving further ,

We can analyse by looking at the dataset rows that there can also be a scope for feature scale bias due to variability in features , ie , wider range and hence improving techniques of normalisation can lead to better results which we have tried applying later on .

Thus , we address the class imbalance bias as well as feature scale bias . Lets look at mitigating them. Bias is affecting the model by biasing it towards specific classes or features. Class imbalance bias causes the model to predict the majority class most of the time and disregard the minority class. Feature scale bias occurs when features with larger ranges dominate the learning process of the model. These biases lower the generalizability of the model and the fairness of the predictions. Their mitigation makes the results more balanced and accurate.

Steps you took to mitigate the bias:

At the mitigation phase, we noticed that the model was biased towards predicting the "Pass" class due to the extreme class imbalance in the data, where the "Pass" students outnumbered the "Fail" students. This imbalance made the model favor the majority class, and it resulted in poor predictions for the minority class. To counteract this, we decided to use SMOTE (Synthetic Minority Over-sampling Technique). We chose SMOTE because it can construct synthetic samples of the minority class in such a way that the dataset becomes balanced without losing any information of the majority class. Through the use of SMOTE, we hoped the model would become more balanced in how it handles both classes and would improve its predictability in both "Pass" and "Fail" students.

Results that we got are :

Initially, when we applied SMOTE with the original normalization technique, we achieved the following results:

Train Accuracy (SMOTE): 0.99

Test Accuracy (SMOTE): 0.76

And the confusion matrix was as follows:

[[22 96]

[0 282]]

This showed that even though the model was extremely precise on the training set, the performance of the model on the test set was better than that we had initially without mitigation but not that upto the mark . It landed at around 76%. More specifically, the confusion matrix demonstrated that there were 96 false positives and only 22 true positives . This meant that the model was still biased to predict the majority class (Pass), but it had correctly identified all the "Fail" students.

However, to try to improve i decided to experiment with a different normalization technique, Min-Max Scaling with SMOTE. The results after applying SMOTE and Min-Max Scaling were:

Train Accuracy: 1.00

Test Accuracy: 0.99

Confusion matrix:

[[116 2]

[0 282]]

Thus, with Min-Max Scaling, the model performed exceptionally well, with very few misclassifications. The test accuracy increased significantly, and the confusion matrix showed a much better balance between the "Pass" and "Fail" predictions. This improvement suggests that Min-Max Scaling helped to better align the features in the same range and scale, making it easier for the model to treat each feature equally.

In doing so, we were able to remove both the class imbalance bias and the feature scaling bias, resulting in absolutely surprising results that greatly enhanced the model's ability to make accurate predictions.

We also tried using random sampling but it led to decrease of test accuracy and hence we did not continue working with it .

Results of comparison

	Train Accuracy	Test Accuracy
Before Mitigation	0.69	0.70
After Mitigation(SMOTE)	0.99	0.76
After Mitigation(SMOTE+Min-Max)	1.00	0.99

Difficulties and issues encountered in the homework:

1.it was very difficult to get an initial start to find bias and detect it for the first time

2. It took a lot of time in comprehension for the case study 2 and to finely get the bias detected .

3. Being very open ended the assignments generalisability was very large and faced challenges while working and learning mitigation techniques .

Conclusion

To conclude ,it was fun learning and actively working with bias detection and mitigation . The task in this assignment was to solve. Through this, we realized the importance of proper data preprocessing and model evaluation techniques in addressing biased datasets, ultimately gaining more equitable and accurate predictions.

Note: Used Ai and chat gpt for concept clarity and working