DS202 Final Project

Haoyu Yang, Wenqing Zhu

May 2023

1 Background

1.1 What is HMDA

Housing mortgage loans are a prevalent type of financial loan in which borrowers use their property as collateral to secure funds from banks or other financial institutions. After borrowers submit their loan applications, banks follow a stringent process to evaluate and approve the application, ensuring the transaction's feasibility for both parties.

1.2 Context

This data is essential for regulatory bodies, policymakers, and researchers to assess whether financial institutions are serving the housing needs of various communities fairly and equitably.

1.3 Content

This dataset encompasses mortgage decisions made in 2015 for the state of New York. Data for additional states and years can be accessed through the Consumer Financial Protection Bureau's website. The dataset includes various attributes, such as applicant demographics, loan details, property information, and the outcome of the loan application.

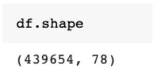
1.4 Purpose

The primary objective of this analysis is to investigate the fairness of mortgage loan decisions across different ethnicities (e.g., Asian, White, Black), genders (male and female), and income levels. We aim to identify potential disparities and biases in lending practices, which can help inform policy recommendations and enhance transparency in the mortgage lending process.

Our Automated Decision System (ADS) evaluates lending decisions based on the information provided by applicants, such as personal demographics (race, gender, income), loan details (amount, purpose), and property characteristics. The system generates an outcome on a scale from 1 to 7, where 1 represents full approval and 7 indicates complete denial. This ADS enables a comprehensive and efficient analysis of mortgage applications, promoting fair lending practices and ensuring compliance with regulatory requirements.

2 Input and output

- The Kaggle link does not provide information on the data sources.
  - We have a 43964 sample size in the data set
  - We have 77 independent variables and 1 dependent variable

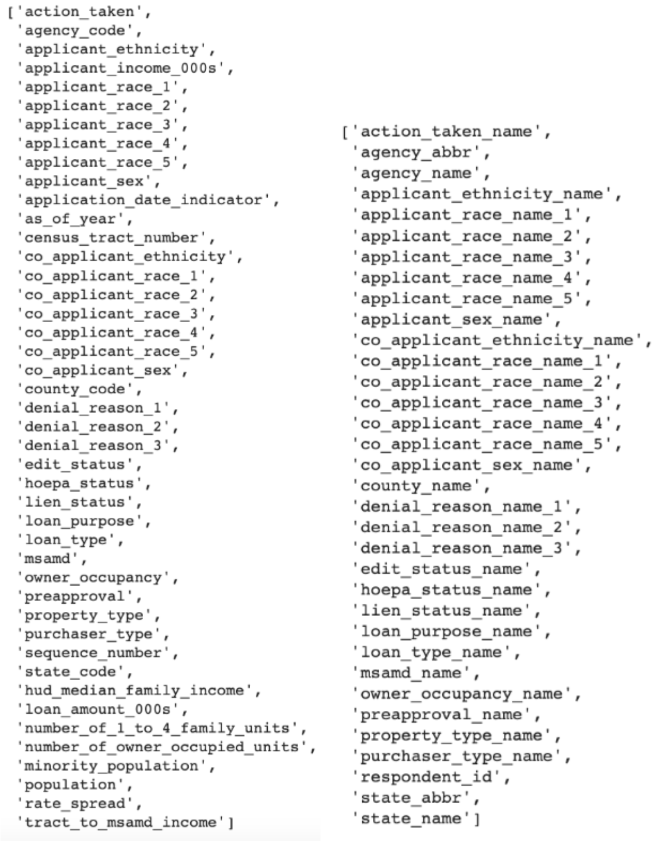


```
df.shape

(439654, 78)
```

Figure 1: Shape of the dataset

- Column types check

  - Numeric columns
  - Categorical columns



```
['action_taken',
 'agency_code',
 'applicant_ethnicity',
 'applicant_income_000s',
 'applicant_race_1',
 'applicant_race_2',
 'applicant_race_3',
 'applicant_race_4',
 'applicant_race_5',
 'applicant_sex',
 'application_date_indicator',
 'as_of_year',
 'census_tract_number',
 'co_applicant_ethnicity',
 'co_applicant_race_1',
 'co_applicant_race_2',
 'co_applicant_race_3',
 'co_applicant_race_4',
 'co_applicant_race_5',
 'co_applicant_sex',
 'county_code',
 'denial_reason_1',
 'denial_reason_2',
 'denial_reason_3',
 'edit_status',
 'hoepa_status',
 'lien_status',
 'loan_purpose',
 'loan_type',
 'msamd',
 'owner_occupancy',
 'preapproval',
 'property_type',
 'purchaser_type',
 'sequence_number',
 'state_code',
 'hud_median_family_income',
 'loan_amount_000s',
 'number_of_1_to_4_family_units',
 'number_of_owner_occupied_units',
 'minority_population',
 'population',
 'rate_spread',
 'tract_to_msamd_income']
```

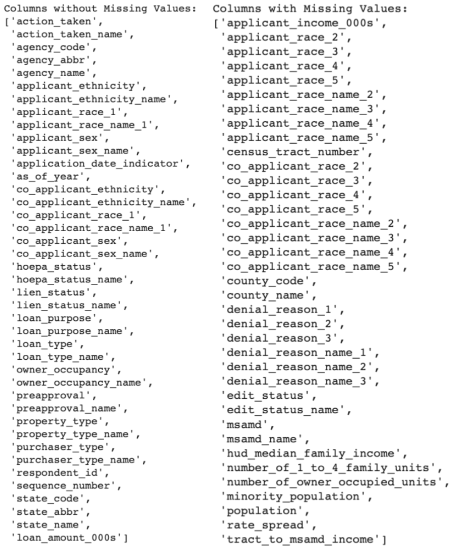
(a)

```
['action_taken_name',
 'agency_abbr',
 'agency_name',
 'applicant_ethnicity_name',
 'applicant_race_name_1',
 'applicant_race_name_2',
 'applicant_race_name_3',
 'applicant_race_name_4',
 'applicant_race_name_5',
 'applicant_sex_name',
 'co_applicant_ethnicity_name',
 'co_applicant_race_name_1',
 'co_applicant_race_name_2',
 'co_applicant_race_name_3',
 'co_applicant_race_name_4',
 'co_applicant_race_name_5',
 'co_applicant_sex_name',
 'county_name',
 'denial_reason_name_1',
 'denial_reason_name_2',
 'denial_reason_name_3',
 'edit_status_name',
 'hoepa_status_name',
 'lien_status_name',
 'loan_purpose_name',
 'loan_type_name',
 'msamd_name',
 'owner_occupancy_name',
 'preapproval_name',
 'property_type_name',
 'purchaser_type_name',
 'respondent_id',
 'state_abbr',
 'state_name']
```

(b)

Figure 2: (a) Numeric columns (b) Categorical columns

- Missing Values Check



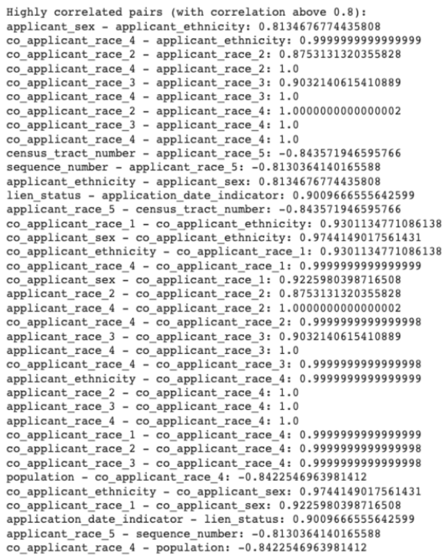
```
Columns without Missing Values:
['action_taken',
 'action_taken_name',
 'agency_code',
 'agency_abbr',
 'agency_name',
 'applicant_ethnicity',
 'applicant_ethnicity_name',
 'applicant_race_1',
 'applicant_race_name_1',
 'applicant_sex',
 'applicant_sex_name',
 'application_date_indicator',
 'as_of_year',
 'co_applicant_ethnicity',
 'co_applicant_ethnicity_name',
 'co_applicant_race_1',
 'co_applicant_race_name_1',
 'co_applicant_sex',
 'co_applicant_sex_name',
 'hoepa_status',
 'hoepa_status_name',
 'lien_status',
 'lien_status_name',
 'loan_purpose',
 'loan_purpose_name',
 'loan_type',
 'loan_type_name',
 'owner_occupancy',
 'owner_occupancy_name',
 'preapproval',
 'preapproval_name',
 'property_type',
 'property_type_name',
 'purchaser_type',
 'purchaser_type_name',
 'respondent_id',
 'sequence_number',
 'state_code',
 'state_abbr',
 'state_name',
 'loan_amount_000s']
```

```
Columns with Missing Values:
['applicant_income_000s',
 'applicant_race_2',
 'applicant_race_3',
 'applicant_race_4',
 'applicant_race_5',
 'applicant_race_name_2',
 'applicant_race_name_3',
 'applicant_race_name_4',
 'applicant_race_name_5',
 'census_tract_number',
 'co_applicant_race_2',
 'co_applicant_race_3',
 'co_applicant_race_4',
 'co_applicant_race_5',
 'co_applicant_race_name_2',
 'co_applicant_race_name_3',
 'co_applicant_race_name_4',
 'co_applicant_race_name_5',
 'county_code',
 'county_name',
 'denial_reason_1',
 'denial_reason_2',
 'denial_reason_3',
 'denial_reason_name_1',
 'denial_reason_name_2',
 'denial_reason_name_3',
 'edit_status',
 'edit_status_name',
 'msamd',
 'msamd_name',
 'hud_median_family_income',
 'number_of_1_to_4_family_units',
 'number_of_owner_occupied_units',
 'minority_population',
 'population',
 'rate_spread',
 'tract_to_msamd_income']
```

Figure 3: Columns with missing values

- Check Correlations between Independent Variables

- We choose numeric variables, using both node representation and heatmap with threshold = 0.8 to show the correlation between independent variables.
- Correlation value:



```
Highly correlated pairs (with correlation above 0.8):
applicant_sex - applicant_ethnicity: 0.8134676774435808
co_applicant_race_4 - applicant_ethnicity: 0.9999999999999999
co_applicant_race_2 - applicant_race_2: 0.8753131320355828
co_applicant_race_4 - applicant_race_2: 1.0
co_applicant_race_3 - applicant_race_3: 0.9032140615410889
co_applicant_race_4 - applicant_race_3: 1.0
co_applicant_race_2 - applicant_race_4: 1.0000000000000002
co_applicant_race_3 - applicant_race_4: 1.0
co_applicant_race_4 - applicant_race_4: 1.0
census_tract_number - applicant_race_5: -0.843571946595766
sequence_number - applicant_race_5: -0.8130364140165588
applicant_ethnicity - applicant_sex: 0.8134676774435808
lien_status - application_date_indicator: 0.9009666555642599
applicant_race_5 - census_tract_number: -0.843571946595766
co_applicant_race_1 - co_applicant_ethnicity: 0.9301134771086138
co_applicant_sex - co_applicant_ethnicity: 0.9744149017561431
co_applicant_ethnicity - co_applicant_race_1: 0.9301134771086138
co_applicant_race_4 - co_applicant_race_1: 0.9999999999999999
co_applicant_sex - co_applicant_race_1: 0.9225980398716508
applicant_race_2 - co_applicant_race_2: 0.8753131320355828
applicant_race_4 - co_applicant_race_2: 1.0000000000000002
co_applicant_race_4 - co_applicant_race_2: 0.9999999999999998
applicant_race_3 - co_applicant_race_3: 0.9032140615410889
applicant_race_4 - co_applicant_race_3: 1.0
co_applicant_race_4 - co_applicant_race_3: 0.9999999999999998
applicant_ethnicity - co_applicant_race_4: 0.9999999999999999
applicant_race_2 - co_applicant_race_4: 1.0
applicant_race_3 - co_applicant_race_4: 1.0
applicant_race_4 - co_applicant_race_4: 1.0
co_applicant_race_1 - co_applicant_race_4: 0.9999999999999999
co_applicant_race_2 - co_applicant_race_4: 0.9999999999999998
co_applicant_race_3 - co_applicant_race_4: 0.9999999999999998
population - co_applicant_race_4: -0.8422546963981412
co_applicant_ethnicity - co_applicant_sex: 0.9744149017561431
co_applicant_race_1 - co_applicant_sex: 0.9225980398716508
application_date_indicator - lien_status: 0.9009666555642599
applicant_race_5 - sequence_number: -0.8130364140165588
co_applicant_race_4 - population: -0.8422546963981412
```

Figure 4: Groups of independent variables with correlation above the threshold

- Node Representation

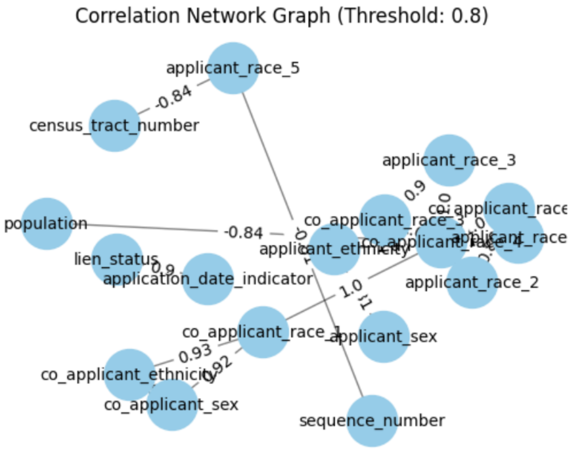


Figure 5: Correlations represented by nodes

- Heatmap

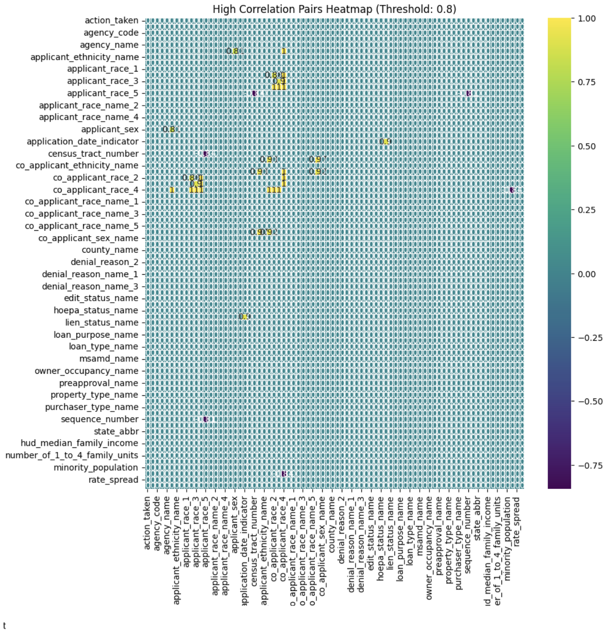
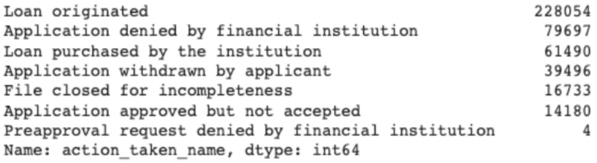


Figure 6: Correlations in heatmap

- Output
  - The output of the system is a score or class label. The y variable ('action_taken') is numeric ranging from 1 to 7. Each number corresponds to a specific action represented in the column named 'action_taken_name' related to the loan application process.
  - Below is the number and corresponding action.
    - 1: Loan originated
    - 2: Application denied by financial institution
    - 3: Loan purchased by the institution
    - 4: Application withdrawn by applicant
    - 5: File closed for incompleteness
    - 6: Application approved but not accepted
    - 7: Pre Approval request denied by financial institution



```
Loan originated                                         228054
Application denied by financial institution              79697
Loan purchased by the institution                        61490
Application withdrawn by applicant                       39496
File closed for incompleteness                           16733
Application approved but not accepted                    14180
Preapproval request denied by financial institution          4
Name: action_taken_name, dtype: int64
```

Figure 7: Value counts for the action takens

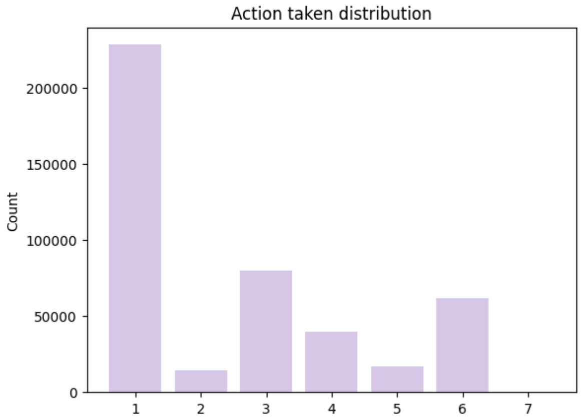
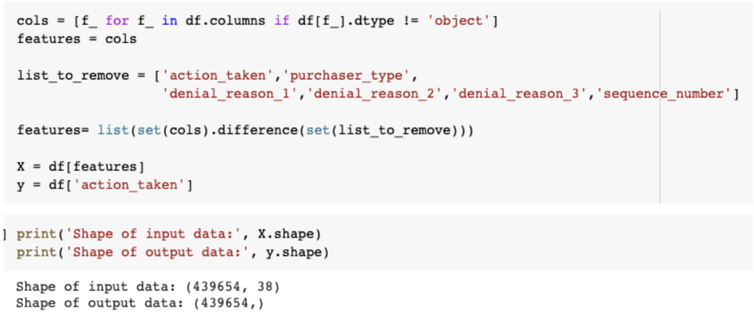


Figure 8: Distribution for action takens

3 Implementation and validation

We will use the data cleaning method from Kaggle competition to pre-process the data set first. In his implementation, the categorical columns are removed from the original dataset. After the modification, there are 38 columns left, including information of applicant sex, race, and income. We will use them to process the ADS. Noticeably, the data pre-processing method doesn't exclude any null values. In our implementation, we are going to replace the null values by the column mean.

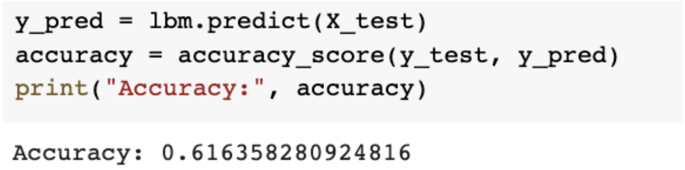


```
cols = [f_ for f_ in df.columns if df[f_].dtype != 'object']
features = cols

list_to_remove = ['action_taken','purchaser_type',
                  'denial_reason_1','denial_reason_2','denial_reason_3','sequence_number']

features= list(set(cols).difference(set(list_to_remove)))

X = df[features]
y = df['action_taken']
```

```
] print('Shape of input data:', X.shape)
  print('Shape of output data:', y.shape)

  Shape of input data: (439654, 38)
  Shape of output data: (439654,)
```

Figure 9: Shape of the dataset after pre-processing

The algorithm we are going to use for prediction is LGBM (Light Gradient Boosting Machine), which is an efficient algorithm to address machine learning problems on large-scale datasets based on the decision tree algorithm. We use the default parameters within the LGBM function. The model accuracy is around 61.6%.



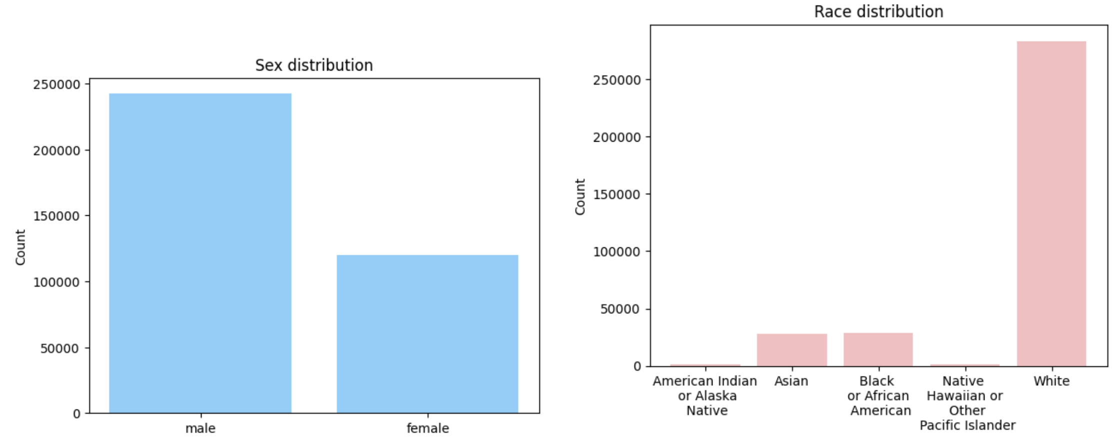
```
y_pred = lbm.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy:", accuracy)

Accuracy: 0.616358280924816
```

Figure 10: Fit the model

4 Outcomes

We will look at the accuracy metrics across

- Subpopulations divided by Sex and Race



(a)

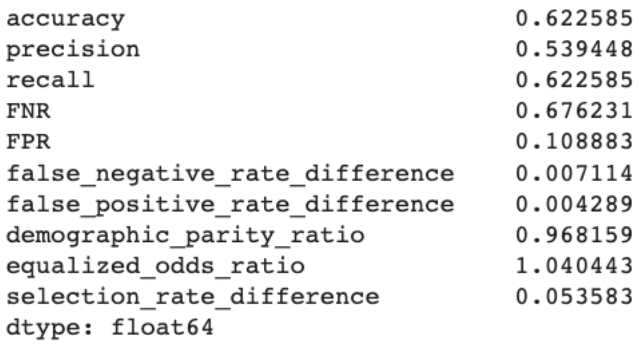
(b)

Figure 11: (a) Distribution of subpopulations divided by sex (b) Distribution of subpopulations divided by race

The accuracy metrics we are going to use include

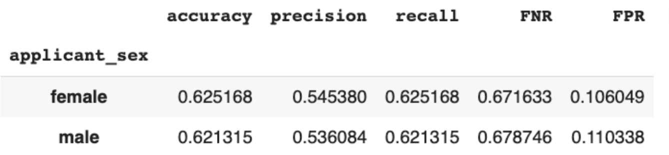
- Accuracy
- Precision
- Recall
- FNR
- FPR
- False negative rate difference
- False positive rate difference
- Demographic parity ratio
- Equalized odds ratio
- Selection rate difference

For subpopulations divided by sex (we will only look at the ‘male’ and ‘female’ groups, for there are null and invalid responses), the result is shown below:



```
accuracy                          0.622585
precision                         0.539448
recall                            0.622585
FNR                               0.676231
FPR                               0.108883
false_negative_rate_difference    0.007114
false_positive_rate_difference    0.004289
demographic_parity_ratio          0.968159
equalized_odds_ratio              1.040443
selection_rate_difference         0.053583
dtype: float64
```

Figure 12: Accuracy metrics for subpopulations divided by sex



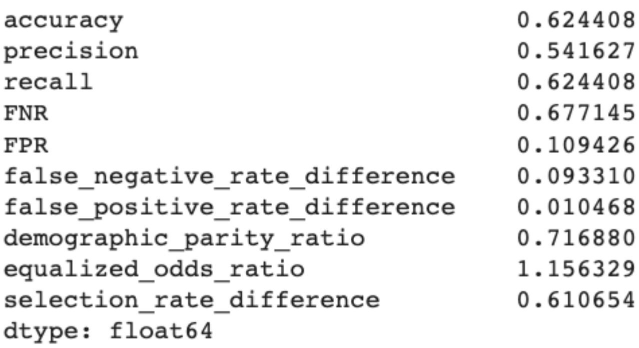
applicant_sex	accuracy	precision	recall	FNR	FPR
female	0.625168	0.545380	0.625168	0.671633	0.106049
male	0.621315	0.536084	0.621315	0.678746	0.110338

Figure 13: Accuracy metrics between subpopulations divided by sex

We can see that:

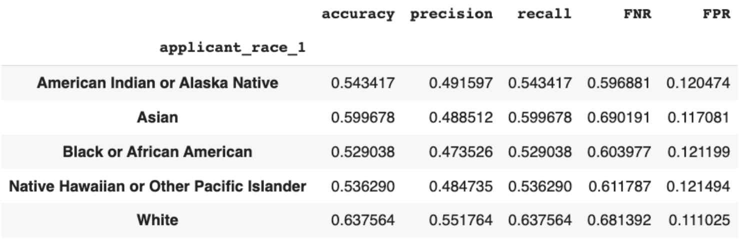
- Overall performance: The accuracy, precision, and recall for both groups are relatively close, with accuracy and recall at around 62% and precision at around 54%. This suggests that the model has a moderate ability to correctly predict sex and identify true positive cases.
- Fairness evaluation: By comparing the false_negative_rate_difference (0.007114) and false_positive_rate_difference (0.004289), it can be inferred that there is a slight disparity between the male and female groups in terms of classification errors. However, these differences are small, suggesting that the model's performance is relatively fair across both groups.
- Demographic parity: The demographic_parity_ratio (0.968159) is close to 1, which implies that the model is almost equally likely to predict positive outcomes for both male and female groups.
- Equalized odds: The equalized_odds_ratio (1.040443) is also close to 1, indicating that the model performs similarly for both demographic groups in terms of classifying true positives and true negatives.
- Separate evaluation: When examining the female and male groups separately, their performance metrics are quite similar. This further supports the notion that the model is reasonably fair in its predictions across both groups.
- In summary, the model demonstrates moderate performance and a reasonable degree of fairness across both male and female subpopulations. The small differences in false_negative_rate_difference and false_positive_rate_difference indicate that there is room for improvement in minimizing disparities between the groups. However, the demographic_parity_ratio and equalized_odds_ratio suggest that the model is relatively fair in its treatment of both male and female groups. To further promote fairness in responsible data science, it is essential to consider these metrics and continuously refine the model to reduce any potential biases.

For subpopulations divided by race (we will only look at the 'American Indian or Alaska Native', 'Asian', 'Black or African American', 'Native Hawaiian or other Pacific Islander' and 'White' groups), the result is shown below:



```
accuracy                          0.624408
precision                         0.541627
recall                            0.624408
FNR                               0.677145
FPR                               0.109426
false_negative_rate_difference    0.093310
false_positive_rate_difference    0.010468
demographic_parity_ratio          0.716880
equalized_odds_ratio              1.156329
selection_rate_difference         0.610654
dtype: float64
```

Figure 14: Accuracy metrics for subpopulations divided by race



	accuracy	precision	recall	FNR	FPR
applicant_race_1					
American Indian or Alaska Native	0.543417	0.491597	0.543417	0.596881	0.120474
Asian	0.599678	0.488512	0.599678	0.690191	0.117081
Black or African American	0.529038	0.473526	0.529038	0.603977	0.121199
Native Hawaiian or Other Pacific Islander	0.536290	0.484735	0.536290	0.611787	0.121494
White	0.637564	0.551764	0.637564	0.681392	0.111025

Figure 15: Accuracy metrics between subpopulations divided by race

We can see that:

- Overall performance: The accuracy, precision, and recall vary across different racial groups, ranging from around 53% to 64% in accuracy and 47% to 55% in precision. This suggests that the model's ability to correctly predict race and identify true positive cases is not consistent across the different racial groups.

- Fairness evaluation: The false_negative_rate_difference (0.093310) and false_positive_rate_difference (0.010468) indicate significant disparities between the racial groups in terms of classification errors. This suggests that the model's performance is not fair across all groups.

- Demographic parity: The demographic_parity_ratio (0.716880) is far from 1, which implies that the model is not equally likely to predict positive outcomes for all racial groups.

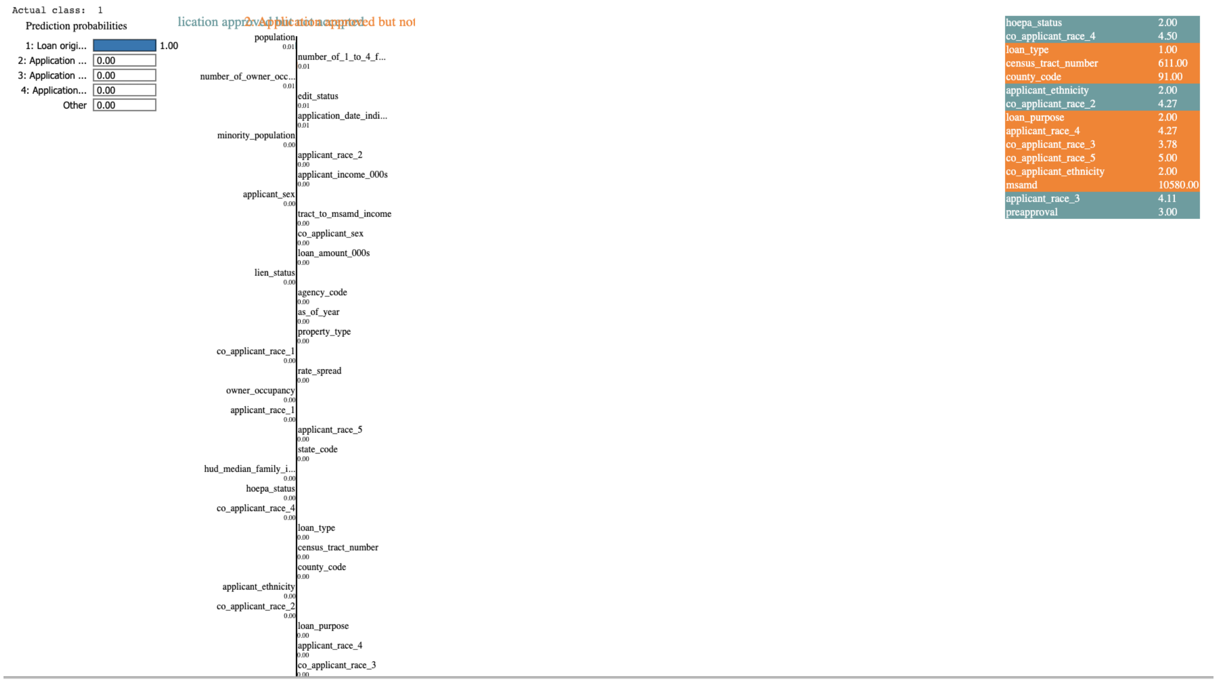
- Equalized odds: The equalized_odds_ratio (1.156329) deviates from 1, indicating that the model performs differently for various demographic groups in terms of classifying true positives and true negatives.

- Separate evaluation: When examining the racial groups separately, the performance metrics differ significantly, suggesting that the model may be biased in its predictions across different racial groups.

- In summary, the model demonstrates inconsistent performance and fairness across the different racial subpopulations. The disparities in false_negative_rate_difference and false_positive_rate_difference, along with the demographic_parity_ratio and equalized_odds_ratio, suggest that the model is not treating all racial groups fairly. To promote fairness in responsible data science, it is essential to consider these metrics and continuously refine the model to reduce any potential biases and ensure equal treatment across all racial groups.

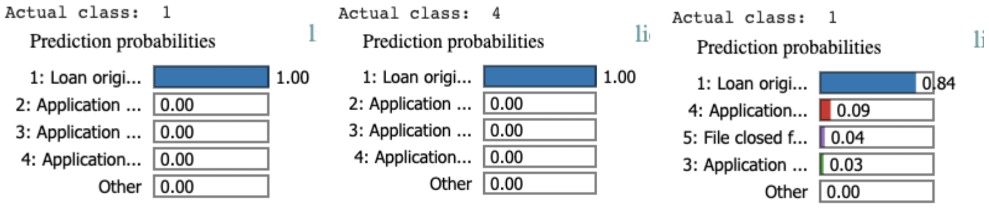
From the results above, we likely see a trade-off between fairness and accuracy: as the accuracy increases, the fairness may be compromised. The ten metrics show how fairness is maintained between the subpopulations (e.g. demographic ratio) as well as how well the model predicts (e.g. accuracy). Overall, the ADS gives good feedback in fairness, but the accuracy can still be improved.

We chose to use LIME to explain individual predictions because it provides a transparent and interpretable way to understand the model's decision-making process. This is particularly important for ADS, as the ability to explain individual predictions can help identify potential biases or errors in the model and improve its overall performance. By using LIME, we can better understand which features are contributing the most to the model's predictions and identify any areas for improvement. Additionally, the visualization provided by LIME can help us communicate the model's performance to stakeholders in a more intuitive and accessible way. We first use SubmodularPick to find out the best indices from the test set to explain the overall performance of the classifier and visualize the results.



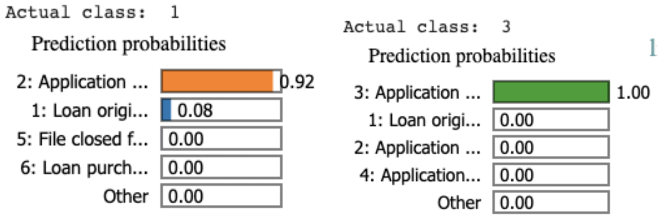
hoepa_status	2.00
co_applicant_race_4	4.50
loan_type	1.00
census_tract_number	611.00
county_code	91.00
applicant_ethnicity	2.00
co_applicant_race_2	4.27
loan_purpose	2.00
applicant_race_4	4.27
co_applicant_race_3	3.78
co_applicant_race_5	5.00
co_applicant_ethnicity	2.00
msamd	10580.00
applicant_race_3	4.11
preapproval	3.00

Figure 16: Shape of the dataset



(a) (b) (c)

Figure 17: (a) Actual class = 1 (b) Actual class = 4 (c) Actual class = 1



(a) (b)

Figure 18: (d) Actual class = 1 (e) Actual class = 3

The result still reflects the relatively poor performance on the accuracy (it successfully predicts 3 out of 5 outcomes). To better train the ADS, we not only should focus on the fairness between subpopulations, but should also pay attention to the model accuracy. To improve the model, we could adjust the number of leaves and the depth of the tree to capture more features. At the same time, we should take care of overfitting by cross-validation.

5 Summary

Overall, we consider the data is relatively appropriate for the ADS because 1) it includes a large scale of training data 2) it includes high dimensions of features. However, the source of the dataset isn't specified and there are too many null values which will affect the outcome to some extent.
The implementation of accuracy metrics is relatively fair. We consider the following metrics to capture the efficiency of ADS:
1. Accuracy: Assesses the classifier's capacity to make accurate predictions for samples.
2. Precision: Evaluates the classifier's accuracy when predicting positive instances.
3. Recall: Determines the classifier's ability to correctly recognize positive instances.
4. False Negative Rate (FNR): Calculates the likelihood of the classifier incorrectly labeling a positive sample as negative.
5. False Positive Rate (FPR): Computes the likelihood of the classifier inaccurately labeling a negative sample as positive.
6. False Negative Rate Difference: Compares the FNR across different groups.
7. False Positive Rate Difference: Contrasts the FPR between various groups.
8. Demographic Parity Ratio: Analyzes the disparity in the percentage of positive predictions among different groups.
9. Equalized Odds Ratio: Investigates the difference in the rate of accurate predictions between distinct groups, given a specific predicted outcome.
10. Selection Rate Difference: Measures the discrepancy in the count of positive predictions between separate groups.
These metrics help us evaluate the performance of the classifier in multiple aspects, including prediction accuracy, bias, and fairness.

The top four important stakeholders in this analysis of mortgage loan decisions are:
1. Applicants: Individuals applying for mortgage loans, directly impacted by lending decisions, and interested in fair evaluation processes.
2. Lenders: Financial institutions providing mortgage loans, responsible for ensuring unbiased and compliant lending practices.
3. Consumer Financial Protection Bureau (CFPB): A government agency enforcing consumer protection laws, promoting fair lending practices, and ensuring financial institutions' compliance.
4. Policy Makers and Regulators: Stakeholders responsible for creating and maintaining a fair financial system, using the analysis to inform policy recommendations and regulate lending practices.
The ADS should still be improved about accuracy and fairness to be used in the public sector.

There are further improvements in the project. Firstly, replacing null values with column means might not be the best implementation for data pre-processing. Some of the groups might benefit from the replaced value while some do not. Simultaneously, we may increase the complexity of the model so that it will capture more features in fitting the dataset with an increase in accuracy.

6 References

Bukun's Notebook/Solution - Machine Learning Explainability Omnibus.
`https://www.kaggle.com/code/ambarish/machine-learning-explainability-omnibus/notebook`

Kaggle Competition Dataset - Home Mortgage Disclosure Act Data, NY, 2015.
`https://www.kaggle.com/datasets/jboysen/ny-home-mortgage`