Income Class Prediction through Expenditure Allocation

Genesis Adam D. Mendoza

About the data

	Total Household Income	Region	Total Food Expenditure	Main Source of Income	Agricultural Household indicator	Bread and Cereals Expenditure	Total Rice Expenditure	Meat Expenditure	Total Fish and marine products Expenditure	Fruit Expenditure	...
0	480332	CAR	117848	Wage/Salaries	0	42140	38300	24676	16806	3325	...
1	198235	CAR	67766	Wage/Salaries	0	17329	13008	17434	11073	2035	...
2	82785	CAR	61609	Wage/Salaries	1	34182	32001	7783	2590	1730	...
3	107589	CAR	78189	Wage/Salaries	0	34030	28659	10914	10812	690	...
4	189322	CAR	94625	Wage/Salaries	0	34820	30167	18391	11309	1395	...

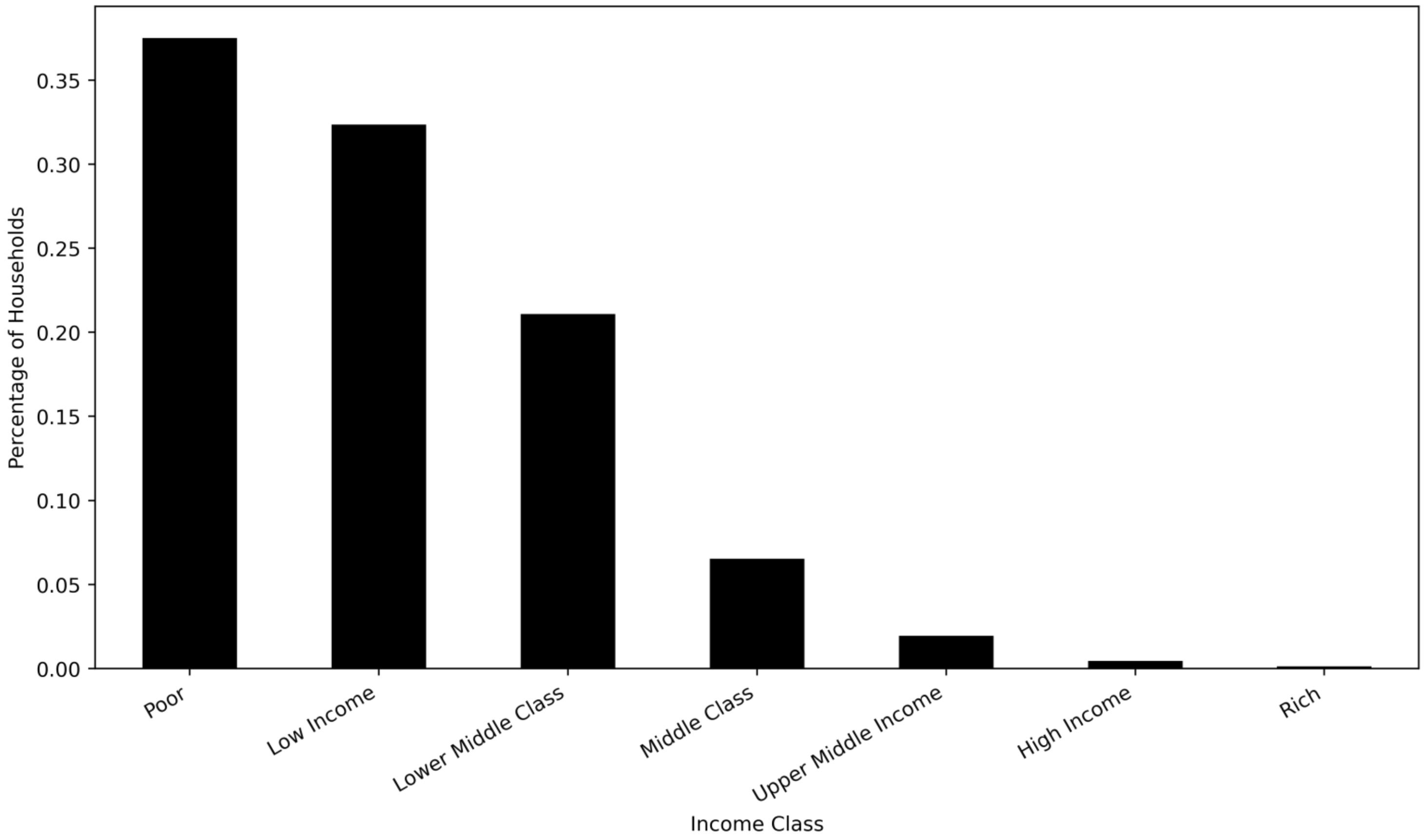
- Family Income and Expenditure Survey (2015) from the Philippine Statistics Authority.
- Includes total annual income, sources of income, items of expenditure, household characteristics, etc.
- Includes $\sim 41{,}500$ households.

Income inequality

- Philippine Institute for Development Studies (PIDS) income classification scheme (monthly)
    - Poor: $[0{,}10957)$
    - Low income: $[10957,\ 21194)$
    - Lower middle class: $[21194,\ 43828)$
    - Middle class: $[43828,\ 76669)$
    - Upper middle class: $[76669, 131484)$
    - High income: $[131484, 219140)$
    - Rich: $[219140, \infty)$
- Does not take family size into account

Income inequality

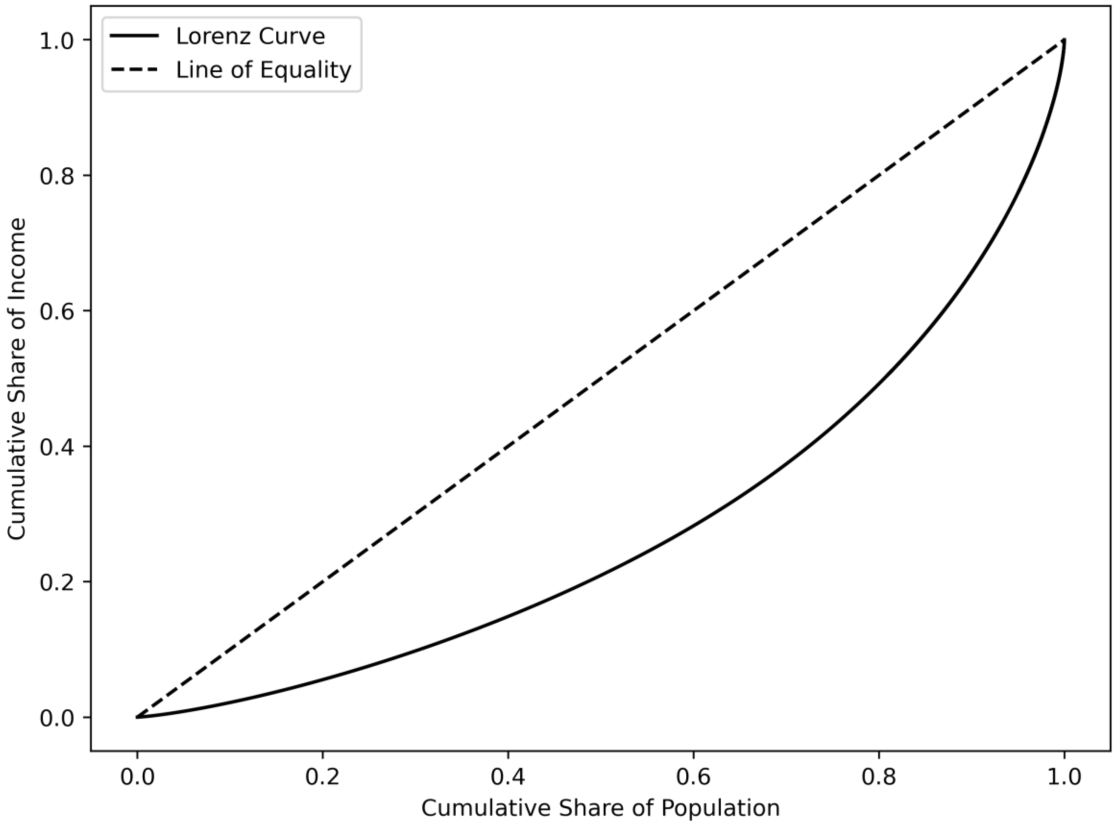
Majority of Filipinos are **poor.**



Philippine Institute for Development Studies (PIDS) income classification scheme

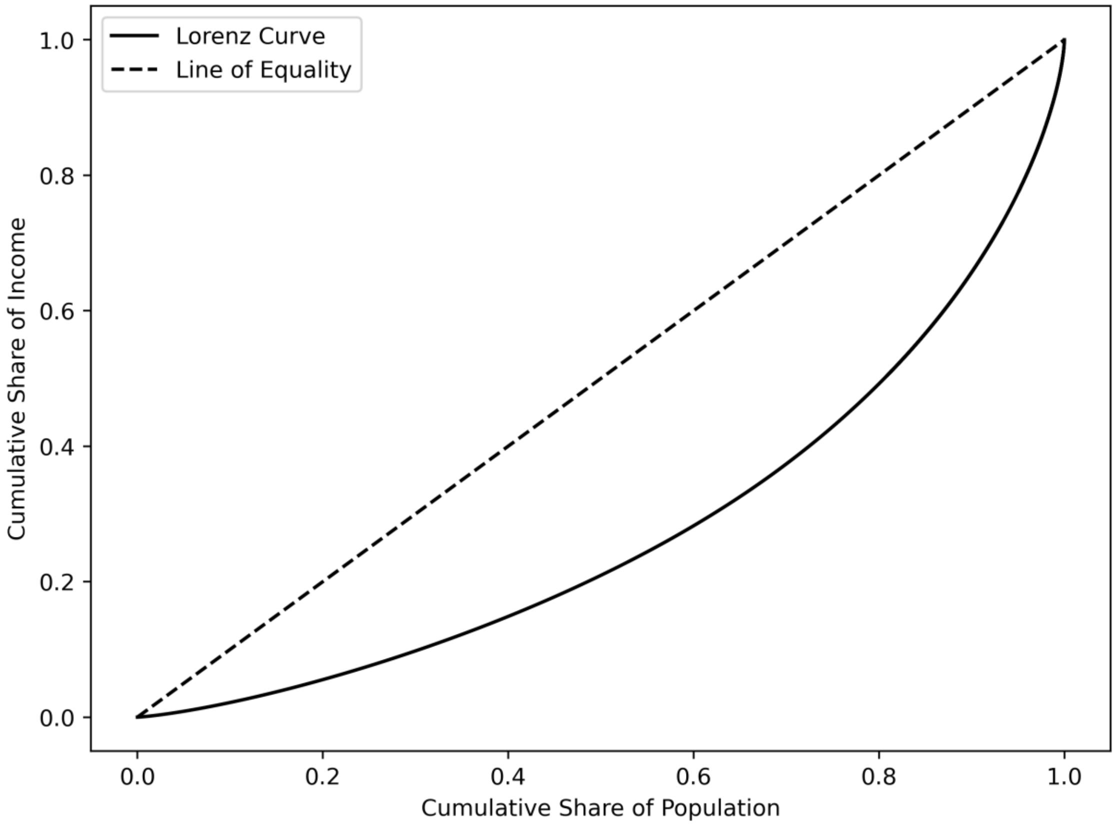
Income inequality

- If we sort the total household income of each family and we take their cumulative sum, we will get its **Lorenz curve**.
- For example, 50% of the population holds just around 20% of the total income.



Income inequality

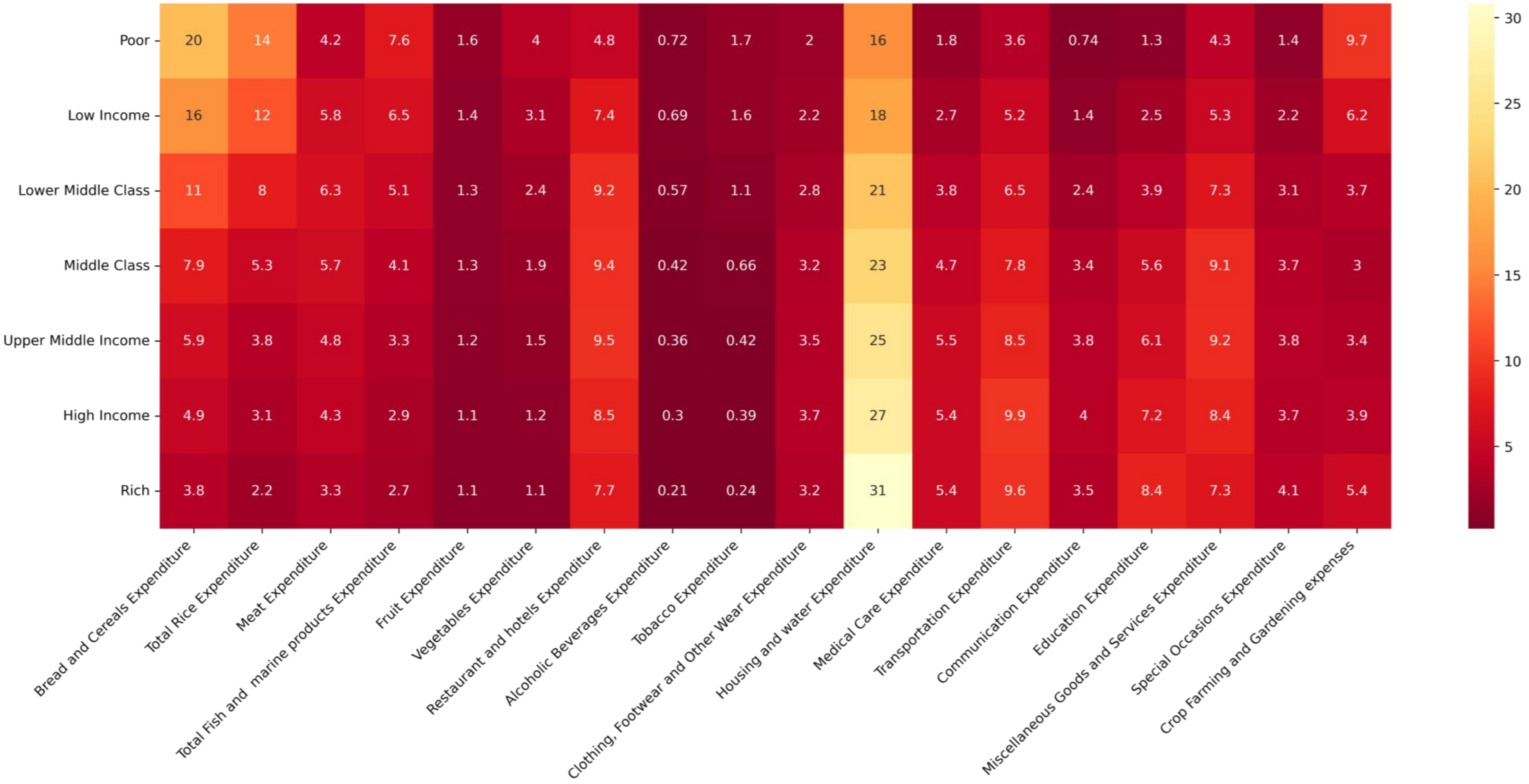
- The Gini coefficient is the area between the line of equality and the Lorenz curve.
- 0 means perfect income equality and 1 means perfect inequality.
- In the FIES (2015) dataset, we have calculated a Gini coefficient of 0.4438.



Expenditure allocation

```python
#Extract the expenditure columns
exp_cols = [col for col in raw_fies.columns if 'Expenditure' in col or 'expense' in col]
tent_feat_cols = exp_cols
fies_tent = raw_fies[tent_feat_cols]

#Remove the redundant column
fies_tent = fies_tent.drop(columns= ['Total Food Expenditure'])
norm_fies = fies_tent.sum(axis = 1)
fies = 100*fies_tent.div(norm_fies, axis = 0)

full = pd.DataFrame(raw_fies['Income Class']).join(fies)
grouped_df = full.groupby('Income Class')[fies.columns].mean()

plt.figure(figsize=(20, 7))
sns.heatmap(grouped_df, annot=True, cmap='YlOrRd_r', linewidths=.5, linecolor='black')
plt.ylabel('Income Class')
plt.xlabel('Expenditure Category')
plt.xticks(rotation=45, ha='right')
plt.savefig('ExpenditureAllocation.png', bbox_inches = 'tight', dpi = 400)
plt.show()
```

Expenditure allocation



Prediction Model

```python
forest_model = RandomForestClassifier(random_state=0)

max_leaf_nodes = [node_val for node_val in range(1, 900, 1)]
max_depth = [depth for depth in range(1, 300, 1)]

random_grid = {'max_leaf_nodes': max_leaf_nodes, 'max_depth': max_depth}
n_estimators = [estim for estim in range(1, 100, 1)]

model_random = RandomizedSearchCV(
    estimator=forest_model,
    param_distributions={**random_grid, 'n_estimators': n_estimators},
    n_iter=30, cv=3, verbose=3, random_state=0, n_jobs=-1
)

pipeline = Pipeline(steps = [('preprocessor', preprocess), ('model', model_random)])
pipeline.fit(x_train, y_train.values.ravel())

opt_estim = model_random.best_params_['n_estimators']
opt_nodes = model_random.best_params_['max_leaf_nodes']
opt_depth = model_random.best_params_['max_depth']
```

```
Fitting 3 folds for each of 30 candidates, totalling 90 fits
The optimal n_estimator is: 99
The optimal max_leaf_nodes is: 747
The optimal max_depth is: 32
The best score for the training data given the optimum parameters is 74.90%
```

- 80 – 20 train-test division
- We use a random forest classifier

Prediction Model

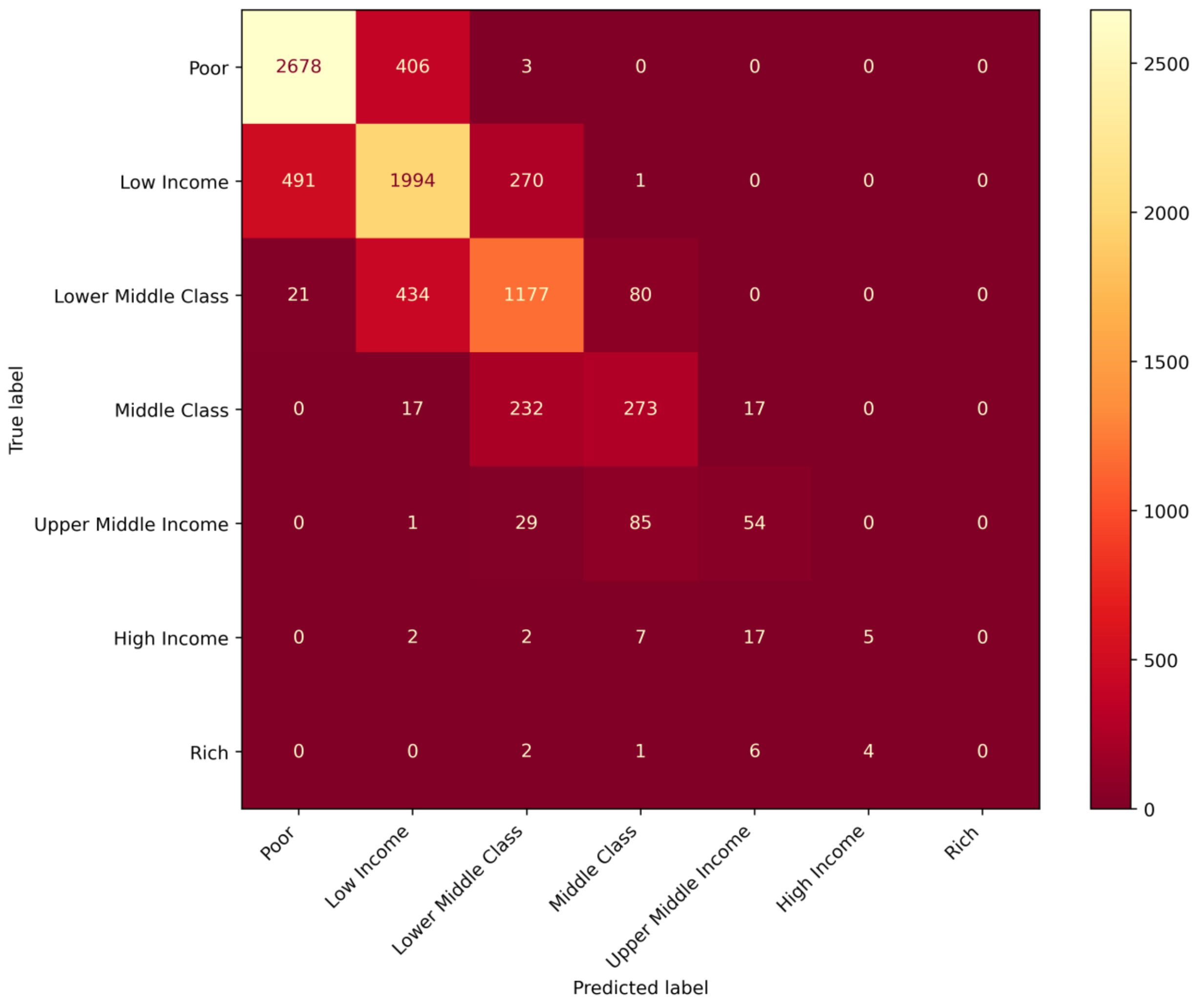
```python
final_model = RandomForestClassifier(random_state=0, n_estimators=opt_estim, max_leaf_nodes=opt_nodes, max_depth=opt_depth)
pipeline = Pipeline(steps = [('preprocessor', preprocess), ('model', final_model)])
pipeline.fit(x_train, y_train.values.ravel())
y_predict = pipeline.predict(x_test)

correct = np.array(y_test == y_predict).sum()/y_test.count()
print('The accuracy of prediction for spending habits is {:.2f}%'.format(correct*100))
```

✓ 9.2s

```
The accuracy of prediction for spending habits is 74.39%
```

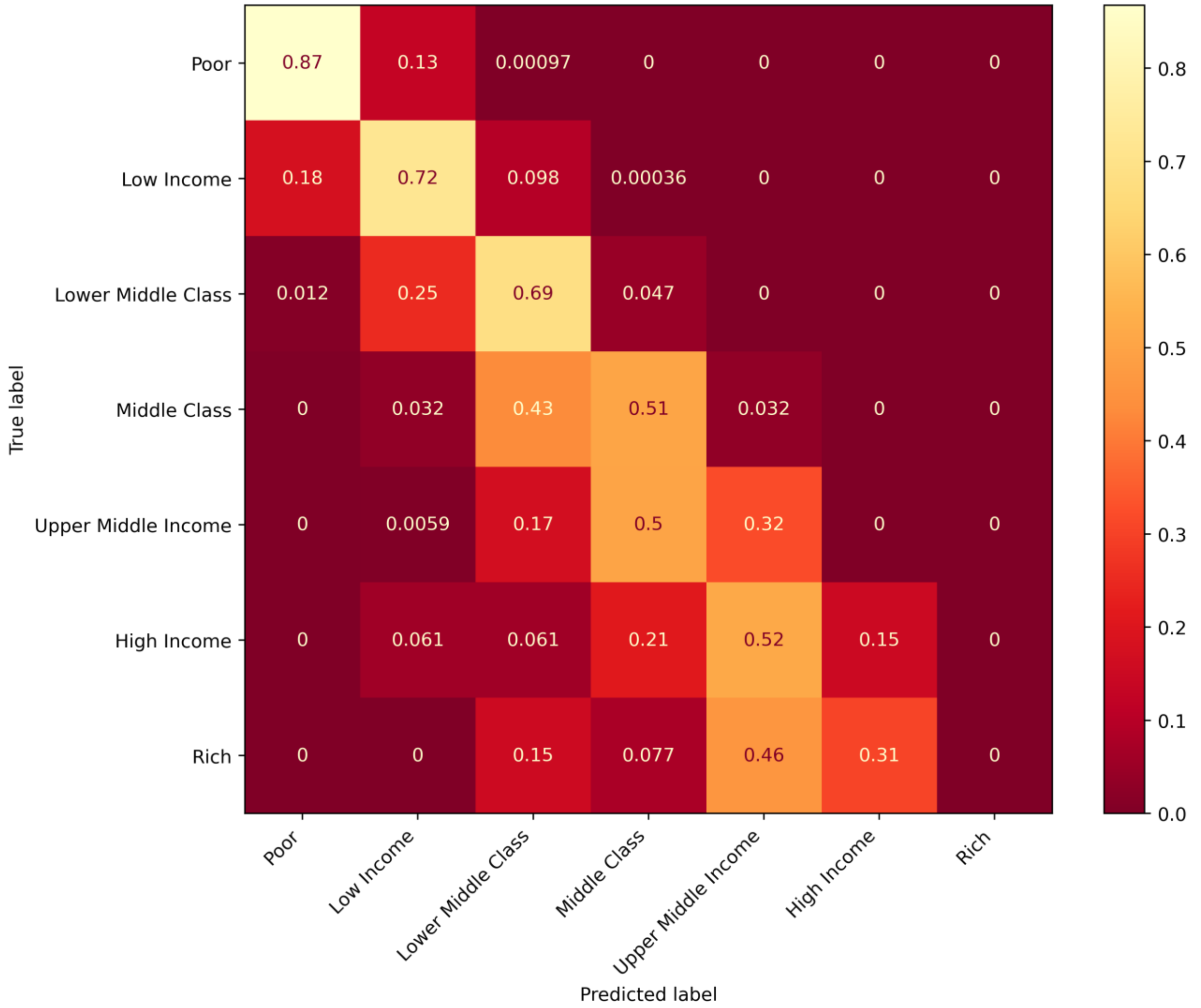
Prediction Model

The model fails to classify higher classes accurately!



Prediction Model

The model fails to classify higher classes accurately!



Synthesis

- The spending behavior of higher income classes are less unpredictable.
- The budget allocation for lower income classes are consistent.
- This may imply that the items with higher expenditure proportions must be the key areas to focus on if we want to address the effects of income inequality.

Bread and Cereals Expenditure	20.369395
Total Rice Expenditure	14.378163
Meat Expenditure	4.232902
Total Fish and marine products Expenditure	7.620689
Fruit Expenditure	1.559692
Vegetables Expenditure	3.954199
Restaurant and hotels Expenditure	4.834301
Alcoholic Beverages Expenditure	0.717937
Tobacco Expenditure	1.715629
Clothing, Footwear and Other Wear Expenditure	1.961966
Housing and water Expenditure	15.861505
Medical Care Expenditure	1.847219
Transportation Expenditure	3.584264
Communication Expenditure	0.744097
Education Expenditure	1.257522
Miscellaneous Goods and Services Expenditure	4.270640
Special Occasions Expenditure	1.402415
Crop Farming and Gardening expenses	9.687467

End