Costa Rican Household Poverty Level Prediction

Project Report by Yuchen Liu and Roy Zhou

Background

A Kaggle competition hosted by the Inter-American Development Bank (IDB) called for an improvement to their income qualification system which they use to allocate financial resources. Social programs often have difficulty ensuring that the right people are given appropriate resources. This is especially challenging in impoverished areas where families lack income and expense records. A popular solution is the Proxy Means Test (PMT), a model that considers a family's observable household attributes to predict the household's level of need. However, as population grows and poverty declines the accuracy of this model has deteriorated.

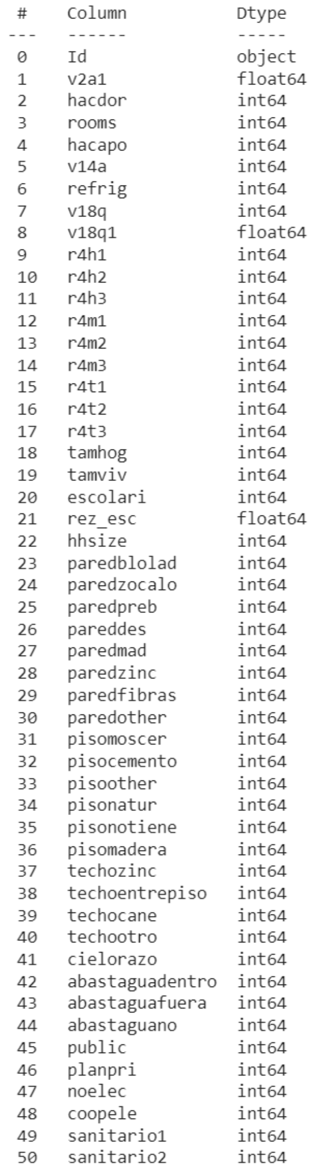
One of the proposed improvements is an automated decision system that employs VotingClassifiers to determine poverty level. We chose this ADS based on its performance in the Kaggle competition and the votes it received from the community. Given the importance and potential impact of the ADS, it is crucial to ensure that the ADS is both accurate and fair.

When evaluating the accuracy and fairness, there are several trade-offs that need to be considered, such as between the false positive rate and true positive rate. While some may argue that it's preferable to occasionally provide aid to a wealthier household to ensure that no poor households miss out, this approach could be incredibly costly and divert resources from those who need it. On the other hand, if the ADS is too strict in its criteria for poverty, many impoverished families may be overlooked. Therefore, there needs to be an appropriate balance to guarantee that we are effectively and equitably distributing resources to those in need.

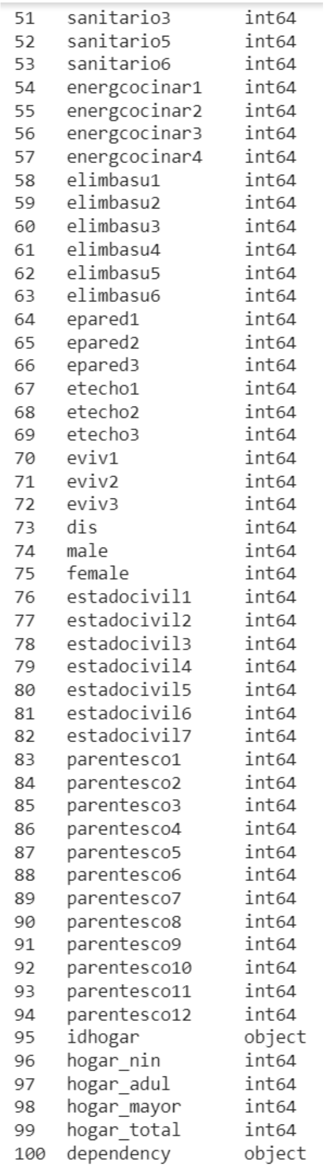
Input and Output

The data was provided directly by the IDB in the form of CSVs. There are 142 columns, with 9,557 rows in the training dataset and 23,856 rows in the validation dataset. Each row represents an individual. Each individual also has a household ID that represents the household they belong to. Other features include house wall material, renting situation, electricity sources, etc. Most of the variables are categorical and are one-hot encoded so they only take values 0 or 1.

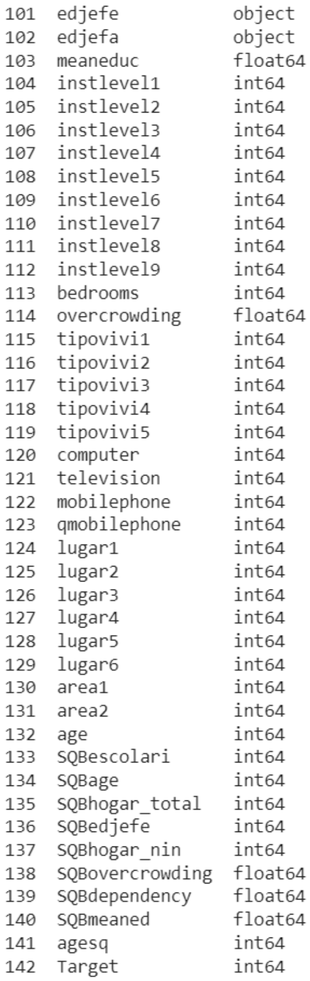
Therefore, most variables are represented by integers or floats. Below is the full list of the variables and their data types:



```
 #    Column           Dtype
---   ------           -----
 0    Id               object
 1    v2a1             float64
 2    hacdor           int64
 3    rooms            int64
 4    hacapo           int64
 5    v14a             int64
 6    refrig           int64
 7    v18q             int64
 8    v18q1            float64
 9    r4h1             int64
 10   r4h2             int64
 11   r4h3             int64
 12   r4m1             int64
 13   r4m2             int64
 14   r4m3             int64
 15   r4t1             int64
 16   r4t2             int64
 17   r4t3             int64
 18   tamhog           int64
 19   tamviv           int64
 20   escolari         int64
 21   rez_esc          float64
 22   hhsize           int64
 23   paredblolad      int64
 24   paredzocalo      int64
 25   paredpreb        int64
 26   pareddes         int64
 27   paredmad         int64
 28   paredzinc        int64
 29   paredfibras      int64
 30   paredother       int64
 31   pisomoscer       int64
 32   pisocemento      int64
 33   pisoother        int64
 34   pisonatur        int64
 35   pisonotiene      int64
 36   pisomadera       int64
 37   techozinc        int64
 38   techoentrepiso   int64
 39   techocane        int64
 40   techootro        int64
 41   cielorazo        int64
 42   abastaguadentro  int64
 43   abastaguafuera   int64
 44   abastaguano      int64
 45   public           int64
 46   planpri          int64
 47   noelec           int64
 48   coopele          int64
 49   sanitario1       int64
 50   sanitario2       int64
```



```
 51   sanitario3       int64
 52   sanitario5       int64
 53   sanitario6       int64
 54   energcocinar1    int64
 55   energcocinar2    int64
 56   energcocinar3    int64
 57   energcocinar4    int64
 58   elimbasu1        int64
 59   elimbasu2        int64
 60   elimbasu3        int64
 61   elimbasu4        int64
 62   elimbasu5        int64
 63   elimbasu6        int64
 64   epared1          int64
 65   epared2          int64
 66   epared3          int64
 67   etecho1          int64
 68   etecho2          int64
 69   etecho3          int64
 70   eviv1            int64
 71   eviv2            int64
 72   eviv3            int64
 73   dis              int64
 74   male             int64
 75   female           int64
 76   estadocivil1     int64
 77   estadocivil2     int64
 78   estadocivil3     int64
 79   estadocivil4     int64
 80   estadocivil5     int64
 81   estadocivil6     int64
 82   estadocivil7     int64
 83   parentesco1      int64
 84   parentesco2      int64
 85   parentesco3      int64
 86   parentesco4      int64
 87   parentesco5      int64
 88   parentesco6      int64
 89   parentesco7      int64
 90   parentesco8      int64
 91   parentesco9      int64
 92   parentesco10     int64
 93   parentesco11     int64
 94   parentesco12     int64
 95   idhogar          object
 96   hogar_nin        int64
 97   hogar_adul       int64
 98   hogar_mayor      int64
 99   hogar_total      int64
 100  dependency       object
```

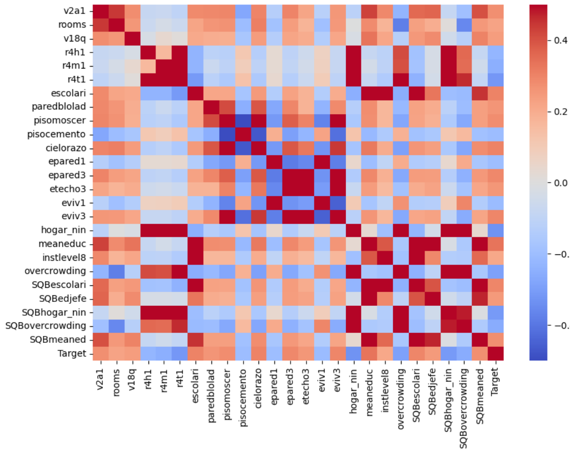


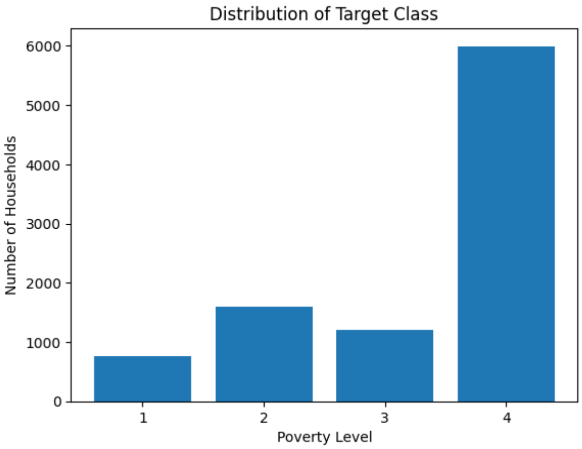
```
 101  edjefe           object
 102  edjefa           object
 103  meaneduc         float64
 104  instlevel1       int64
 105  instlevel2       int64
 106  instlevel3       int64
 107  instlevel4       int64
 108  instlevel5       int64
 109  instlevel6       int64
 110  instlevel7       int64
 111  instlevel8       int64
 112  instlevel9       int64
 113  bedrooms         int64
 114  overcrowding     float64
 115  tipovivi1        int64
 116  tipovivi2        int64
 117  tipovivi3        int64
 118  tipovivi4        int64
 119  tipovivi5        int64
 120  computer         int64
 121  television       int64
 122  mobilephone      int64
 123  qmobilephone     int64
 124  lugar1           int64
 125  lugar2           int64
 126  lugar3           int64
 127  lugar4           int64
 128  lugar5           int64
 129  lugar6           int64
 130  area1            int64
 131  area2            int64
 132  age              int64
 133  SQBescolari      int64
 134  SQBage           int64
 135  SQBhogar_total   int64
 136  SQBedjefe        int64
 137  SQBhogar_nin     int64
 138  SQBovercrowding  float64
 139  SQBdependency    float64
 140  SQBmeaned        float64
 141  agesq            int64
 142  Target           int64
```

The relationships between the features can be visualized using a heatmap, which shows the magnitude and direction of the relationship between features using a gradient color scale. Because we have 141 features, we created a heatmap of only the most correlated features. In general, we found that features are very weakly correlated with one another. The features that are highly correlated with one another are usually the result of the one-hot encoding. For example, there are four features related to where the household gets its electricity from which are all highly correlated. The heatmap can be found below on the left.

Helper functions from ydata-profiling were used to better understand the relationships between features. We found that certain features had really high correlations, such as v2a1 (monthly rent) and pisoother (material of the floor), and r4h3 (total males) and r4h2 (males ages 12 and under). Some of this information might be redundant, but it could be helpful in gauging the ratio of children to adults in a household, and also give a rough idea of the ages of the individuals of a household. We also found some features that were heavily imbalanced. For instance, v14a (whether the household had a bathroom) was made up of 99% 1s, meaning that almost no households lacked a bathroom. This was also the case with paredfibras (whether the predominant material of walls was natural fibers).

The target class is the poverty level of a household. Specifically, the model outputted a number 1 through 4, with 1 indicating extreme poverty and 4 indicating a non-vulnerable household. Households labeled 1, 2, or 3 suggest that a certain amount of aid is needed while households labeled non-vulnerable households shouldn't receive any aid. Interestingly, our training dataset is extremely imbalanced, with more households labeled 4 than households labeled 1, 2, and 3 combined. This disparity can be seen in the bar graph below on the right. The figure on the left is the heatmap showing relationships between the most correlated features.

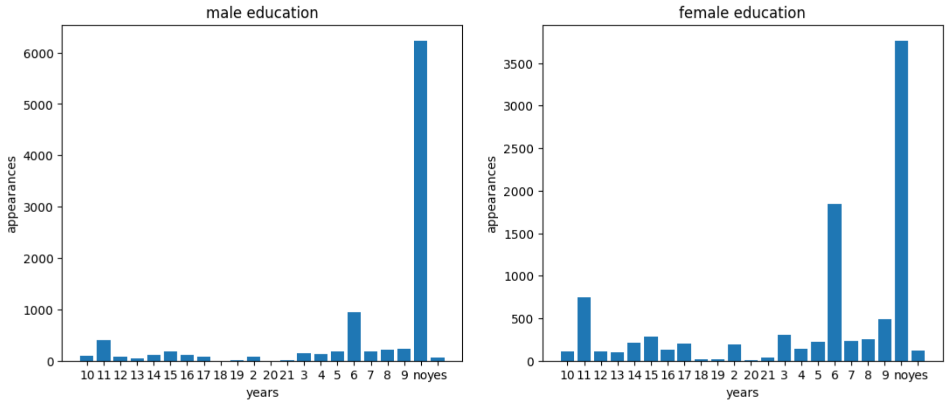




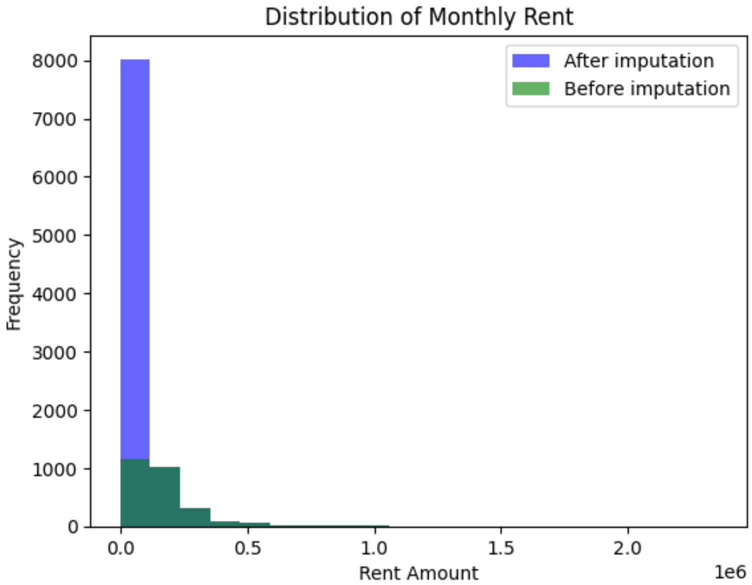
Implementation and Validation

The first step in the ADS was to clean and preprocess it. This involved converting objects to numeric types. Only a handful of features are represented with objects rather than integers or floats. One example is edjefe/edjefa, which represents the years of education for the male and female heads of the household respectively, which should be represented by a number, but because many people replied “yes” or “no”, this feature is represented by a string. In addition, inconsistencies in the data were attempted to be fixed. For example, separate individuals from the same household would answer differently whether or not the household has a bathroom. These inconsistencies were fixed by making conclusions based on the other features. So if the household does not have running water, it’s safe to assume that they also do not have toilets. Preprocessing the data included the following steps:

- Creating new columns to represent relationships between other columns i.e. 'working_man_fraction" was created by (working males) / (total persons in household). A potential concern with this method is discrimination on sensitive features, namely gender. The dataset lacks data about women in general, so this method groups working women, non-working women, and children all into the same group, while not doing so to men.
- Feature transformation by converting one-hot encoded columns into label encoding. A potential issue with this is the introduction of ordinal relationships between categorical variables. Larger values might weigh more in the model and can therefore add bias.
- In the case of years of education for the male and female heads of household, "yes" was replaced with value 4. This method of imputation discriminates against women who on average had more years of education (8.78 vs. 8.48). Since women have a higher average, replacing missing values with 4 for both groups is unfair for women.



- Missing monthly rent values were replaced with 0. This is extremely problematic because it drastically changes the distribution of monthly rent and makes the heavy assumption that missing rent data means no rent payment. In total, over 4000 samples were affected, and the mean monthly rent went from ₡165,231.61 before imputation to ₡46,628.61. This could possibly have a huge impact on predictions made by the ADS.



- Some columns were aggregated to remove redundancy and create more useful features. For example, the number of rooms per person in the household might be a better indicator of poverty level than looking at the number of rooms and the number of people in the household separately. It's not clear how the importance of these features was determined since the author says that they were taken from another kernel.
- The final step was to group the data by household to avoid leakage since individuals that belong to the same household would end up getting the same target label.
- After the preprocessing, we observed that some variables were correlated with sensitive features. For example, parentesco2 (whether the individual has a spouse) was correlated with gender and estadocivil (marital status) was correlated with age. If we removed sensitive features, the model could still discriminate based on these correlated variables.

VotingClassifier models were used to make predictions on the poverty level. These models work by combining the predictions from multiple individual models to make a prediction. The predictions that received the most "votes" would be used. The models also implement early stopping, a regularization technique to prevent overfitting. The idea behind early stopping is that as the model trains on the dataset, its performance may initially increase before decreasing after a certain point due to the model learning noise in the data. Early stopping stops the training process at that point so that the model has learned the general patterns but not the noise. The performance metric used to determine when to stop training is macro-average F1 score, which is the average F1 score for each of the models.The VotingClassifiers also employ Light Gradient Boosting Machine (LGBM), a tree-based learning algorithm that builds a set of decision trees with each one trying to correct the errors of the previous. The ADS uses two separate VotingClassifiers, one made up of XGBClassifiers and another made up of RandomForests. Final predictions of poverty level are determined by combining the votes of the models.

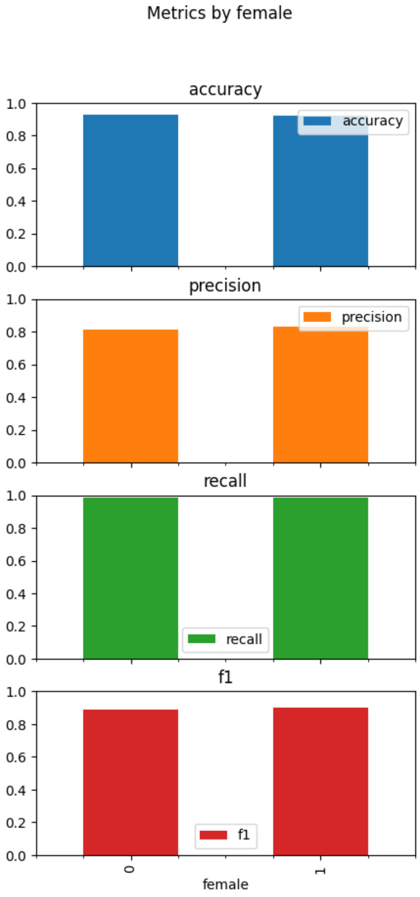
To validate the predictions made by the ADS, predictions can be made on test data that was previously set aside. This is a technique called cross validation. This solution also received the highest score out of over 600 other entries. Therefore its predictions were the most accurate among the participants in the Kaggle competition, meaning the ADS fulfilled its stated goals of predicting the poverty level of households in Costa Rica.

Outcomes

The model performed very well in terms of accuracy metrics. The accuracy metric we looked at was accuracy, which represents the proportion of predictions that the model got right. The model had an accuracy of 0.9258, meaning that around 91% of the poverty level predictions that it made were correct. We also decided to take a look at precision and recall. Precision measures the proportion of correctly predicted positive instances and should be optimized when false positives are costly. In this situation, if the model were to give aid to a household that did not need it, valuable resources would be taken away from those who do. Recall measures the proportion of correctly predicted positive instances out of all actual positive instances and should be optimized when false negatives are costly. In this situation, if the model were to incorrectly classify a poor household as non-vulnerable, then they would be missing out on financial support that they need. Since both of these situations are not ideal, it is important that the model balances precision and recall. The model had a recall score of 0.9861, meaning that it was able to correctly identify around 95% of the households that needed financial support. The model also had a decent precision score of 0.8208, meaning that around 86% of the households that were labeled as needing aid actually needed it. Another metric we looked at was F1 score, which balances precision and recall and it is ideal for imbalanced datasets like ours. The model had an F1 score of 0.9122, which suggests that overall it achieved a good balance between precision and recall with a high overall accuracy.

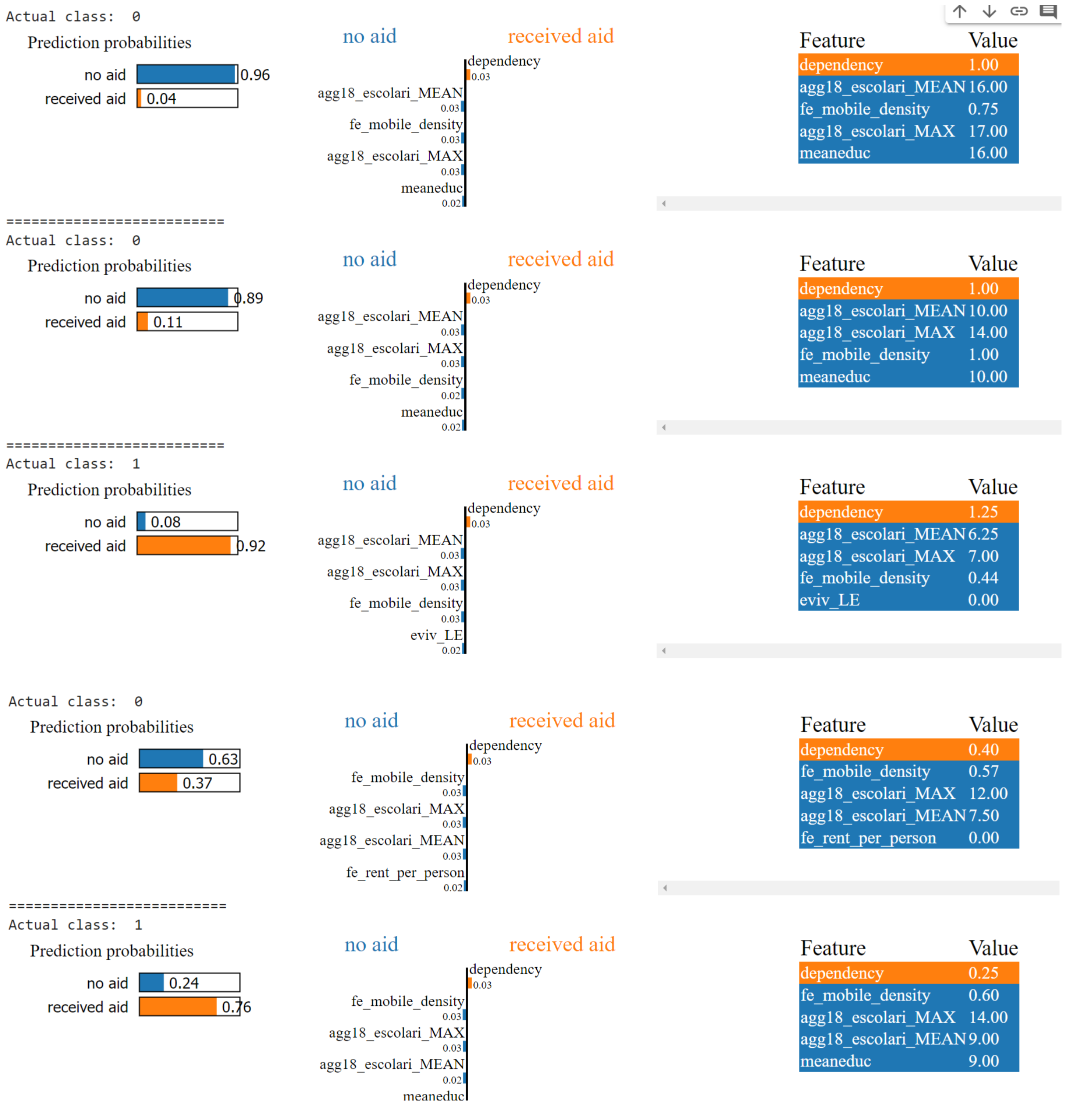
It also seems like the model is not more or less accurate based on gender, one of the protected features that we are concerned with. If we take a look at the same four accuracy metrics from before (accuracy, precision, recall, and F1 score) and split them based on gender, we see that the model achieves the same score regardless of gender. This is visualized by the bar plots on the right. To further determine if the model is discriminating based on gender, we can take a look at fairness metrics such as false negative rate difference and false positive rate difference. These two metrics measure the difference in error rates between different groups. The model received a value of 0.003 for both metrics, which suggest that its tendency to generate false positives or false negatives is consistent across genders. Therefore, the model does not disproportionately

misclassify based on gender. It means that group fairness is satisfied because we have equal outcomes across groups.



The next fairness metric we looked at was demographic parity ratio, which measures the ratio between the selection rates for each gender. The closer the value is to 1, the more equitable the distribution of positive predictions is. The model received a demographic parity ratio of 0.897, which indicates a slight difference in selection rates between the two groups, but it does not necessarily imply a significant bias. We also took a look at equalized odds ratio, which measures how similar the TPR and FPR are across gender groups. Again, a value close to 1 indicates more equitable predictions being made by the model. The model had an equalized odds ratio of 0.974, which suggests a small difference in TPR and FPR rates access groups. This again implies that the model's performance is consistent with respect to gender. The final fairness metric we looked at was selection rate difference, which measures the disparity in the positive prediction rates between the groups. In this scenario it is looking for the difference between the proportion of men who are classified as needing aid compared to the proportion of women. The model received a difference of 0.046, a small difference, which again suggests that there is only a small variation in the model's propensity to predict positive outcomes between genders. Based on these fairness metrics, it seems that the ADS does not exhibit discrimination based on gender. The small differences in FNR rates, FPR rates, selection rates, and the demographic parity ratio all point towards relatively equitable treatment of the groups.

Finally, to better understand the process behind how predictions were made, we used Local Interpretable Model-agnostic Explanations (LIME) to generate explanations for the predictions. This provided for interpretability of the ADS by making the complex models of the black-box machine learning model easier to understand. It also provided insights into how the ADS arrived at a particular prediction and allows us to identify important features in the model. Below are the explanations of predictions that were determined to be indicative of the decision making process:



```
Actual class:  0
```

Feature	Value
dependency	1.00
agg18_escolari_MEAN	16.00
fe_mobile_density	0.75
agg18_escolari_MAX	17.00
meaneduc	16.00

```
=========================
Actual class:  0
```

Feature	Value
dependency	1.00
agg18_escolari_MEAN	10.00
agg18_escolari_MAX	14.00
fe_mobile_density	1.00
meaneduc	10.00

```
=========================
Actual class:  1
```

Feature	Value
dependency	1.25
agg18_escolari_MEAN	6.25
agg18_escolari_MAX	7.00
fe_mobile_density	0.44
eviv_LE	0.00

```
Actual class:  0
```

Feature	Value
dependency	0.40
fe_mobile_density	0.57
agg18_escolari_MAX	12.00
agg18_escolari_MEAN	7.50
fe_rent_per_person	0.00

```
=========================
Actual class:  1
```

Feature	Value
dependency	0.25
fe_mobile_density	0.60
agg18_escolari_MAX	14.00
agg18_escolari_MEAN	9.00
meaneduc	9.00

The LIME explanations reveal possible sources of discrimination based on gender. Many of the features that appear often, such as agg18_escolari_MEAN, agg18_escolari_MAX, and meaneduc, are all related to education. We mentioned previously that females have a slightly higher mean number of years of education than men, and because the ADS's prediction is affected by years of education, it might inadvertently disadvantage females. However, the bigger problem lies in feature fe_mobile_density, which measures the proportion of females in a household with a mobile phone. Based on the LIME explanation, this feature points the prediction towards the household not receiving aid. Essentially, the model is using the number of females in a household who own a mobile phone to determine whether or not the household is eligible for financial assistance. This is highly problematic, especially because it does not do the same thing for males.

Summary

The data included a rich set of information that provided a comprehensive picture of a family's socioeconomic status. The data includes living conditions and family relationships. For the most part, data is well-formed and has enough information to satisfy our goal of determining financial aid distribution. Therefore, data is appropriate for our use case.

Overall, ADS has a high accuracy of 0.92. However, it has significantly lower recall (0.82) and precision (0.82). This means that the model is more likely to be false positives and false negatives, suggesting that model is more likely to give financial aid to people who's not in need. In accuracy metrics across subpopulations, the model maintains the same performance metrics across groups and therefore guarantees group fairness. In fairness metrics, it shows that our model is treating females and male equally. A small selection difference of 0.029 suggests that both male and female have equal chances of being given financial aid. However, the demographic parity ratio of 0.78 and equalized odds ratio of 0.77 are less than ideal; it suggests that one group is less favorable than the other.

We wouldn't recommend deploying this model into production. Even though the ADS has high accuracy metrics across groups and solid fairness metrics, we can't ignore the fact that it uses sensitive features like gender and fraction of females owning mobile phones in households. While ADS may perform well on current validation dataset, we can't guarantee that the model won't discriminate based on sensitive features including gender. When deployed into the real world, there might be model drifts that create significant prediction differences between each subgroup.

First, we recommend avoiding certain aggregations in pre-processing. In feature extraction steps, we should aggregate new features for both genders, and not just for male. In data imputation, we need to consider potential bias introduced by each imputed value; we should consider options including 0, mean, or median and choose the methods with least bias. Before input to ADS, there're still features that correlate with sensitive columns like gender. We can use a correlation remover to prevent models from using sensitive features. In the prediction model, we should rely on XGBoost's feature importance as the sole indicator. To be able to explain features without knowing specific models and give detailed explanations on a particular instance, we should employ LIME to show a better picture. Therefore, on top of specific model's feature importance, we should add LIME to increase interpretability and accountability of ADS.

Sources

Original Kaggle Competition:
https://www.kaggle.com/competitions/costa-rican-household-poverty-prediction

Automated Decision System:

https://www.kaggle.com/code/skooch/xgboost