Dropping columns

1. Dropping constant columns
2. Dropping index column
3. Dropping columns with >50% value missing (NaN or blanks)
4. Dropping columns with more than 99% values in one category
5. Dropping one column from a pair of highly correlated columns (>0.90)
6. Dropping one of two exactly same columns
7. Dropping one columns from a pair of two similar ones delivering similar information and having much lower predictive power with target label

Feature Engineering

1. Converting highly skewed continuous columns(>90%)having most values as 0 into binary feature because of low predictive power, else log transformation
2. Separating categorical and numerical columns-
   Categorical-
   - Object type
   - Discrete numerical with no order

   Numerical-
   - Number type
   - Binary(categorical binary converted to 0/1 features)
3. Combining unexplained categories into mode category
4. Checking correlation of categorical features with target label, dropping columns containing missing values and having low correlation or redundant columns before imputing
5. Replacing blank/ambiguous missing values with np.nan
6. Outlier handling- Capping at 99%
7. Imputation- Numerical- Median, Categorical- Mode
8. Encoder- Target encoder for categorical
9. Scaler- Min Max scaler
10. Baseline model- logistic regression
11. Other techniques- SMOTE, RF classifier, XGB classifier,catboost,

Changes and impacts on metrics

1. Imputation strategy- instead of mode, imputed low cardinality columns with mode and high cardinality ones with ‘Missing’ category -> Increased a little ROC-AUC
2. Frequency encoding instead of target encoding- Increased ROC AUC from 0.58 to 0.63, but reduced the precision-recall of minority category to 0.02 , not a good strategy
3. Standard scaler instead of minmax scaler- No change

4. Using One Hot Encoding for Low cardinality columns and Target encoding for high cardinality columns-> Improved ROC-AUC, precision, recall across all models
5. Using select k best/ xgb model based for feature selection- Improved score
6. Voting classifier improved accuracy- best roc-aic score till now
7. Cross validation helped identify overfitting of model. Stratified CV was used to ensure proper representation of subgroups
8. SHAP was used for model interpretability.
9. Fairlearn used to identify demographic features that had bias. Like- RESIDENCIAL_STATE, STATE_OF_BIRTH
10. Bias mitigation strategies - build a simple bias dashboard for suspected demographic features including precision, recall, selection_rate and demographic parity difference.
11. Appropriate techniques such as SMOTE and boosting models were used to reduce bias.