

# Business Solution

AUTHOR

Varun Selvam

## Load Libraries

```
library(tidyverse) # Load the tidyverse library
```

## Import the Datasets

```
#pred = predictions
# Read in predictions from the XGboost model
XGb_boost_pred <- read_csv("C:\\Users\\User\\Box Sync\\Business Analytics Degree\\Semesters\\Fall 2020\\XGBoost Predictions.csv")

# Read in cleaned dataset that was used to train and cross-validate models
train_dataset <- read_csv("C:\\Users\\User\\Box Sync\\Business Analytics Degree\\Semesters\\Fall 2020\\Cleaned Dataset.csv")
```

## Financial Impact

### Predicting with Majority Classifier

```
# Create a column that contains all the predictions for a majority classifier. # This will be zero
train_dataset <- train_dataset %>%
  mutate(majority_classifier_pred = 0)

# 0 represents positive class, 1 represents negative class
train_dataset <- train_dataset %>% # Overwrite df with changes
  mutate( # mcc = majority classifier comparison
    MCC = case_when(
      TARGET == 0 & majority_classifier_pred == 0 ~ "correct loan approval",
      TARGET == 1 & majority_classifier_pred == 0 ~ "incorrect loan approval",
      TRUE ~ NA_character_ # Default case for unexpected values
    )
  )
```

The code above adds a column that contains all of the Majority Classifier's predictions. Since the majority class is zero, this means that all the predictions for a Majority Classifier will be zero.

# Predicting with XGboost Model

```
# Combine the XGBoost predictions with the train dataset and assign to combined_df.
combined_df <- cbind(XGb_boost_pred,train_dataset)

combined_df <- combined_df %>% # Overwrite dataframe with the new changes
  mutate( # XGbc = XGboost comparison
    XGbc = case_when(
      TARGET == 0 & XGb_boost_pred == 0 ~ "correct loan approval",
      TARGET == 1 & XGb_boost_pred == 1 ~ "true negative",
      TARGET == 0 & XGb_boost_pred == 1 ~ "false negative",
      TARGET == 1 & XGb_boost_pred == 0 ~ "incorrect loan approval",
    )
  )
```

The code above adds a column that compares Xgboost's predictions to the target column and identifies whether the prediction matches or not. If the results match, then the code will return "correct loan approval" for positive classes and "true negative" for the negative class.

Afterwards, if the target value is 0 but the prediction value was 1, then the code will return false negative because it predicted the negative class when it should have predicted the positive class.

Likewise, if the target value is 1 but the prediction value was 0, then the code will return "incorrect loan approval" because it indicates that Xgboost classified/approved a loan when it should have classified the loan as negative.

## Comparison of Predictions

The below code compares how the majority classifier and Xgboost model compare against correctly predicting loans or incorrectly predicting loans.

```
combined_df %>%
  group_by(MCC) %>% # Group by majority classifier comparison column
  summarise(median_amt = median(AMT_ANNUITY), # Calculate the median annuity amount
    n = n()) # Get a count of the number for each category
```

# A tibble: 2 × 3

MCC	median_amt	n
<chr>	<dbl>	<int>
1 correct loan approval	23800.	218531
2 incorrect loan approval	24412.	19245

```
combined_df %>% # Group by majority classifier comparison column
  filter(XGbc == "correct loan approval" | XGbc == "incorrect loan approval") %>% # Filter to only
  group_by(XGbc) %>% # Sort results by XGboost Comparison
```

```
summarise(median_amt = median(AMT_ANNUITY), # Calculate the median annuity amount
          n = n()) # Get a count of the number for each category.
```

# A tibble: 2 × 3

	XGbc	median_amt	n
	<chr>	<dbl>	<int>
1	correct loan approval	23800.5	217980
2	incorrect loan approval	24363	17940

- mcc = majority classifier comparison
- XGbc = XGboost comparison

When a loan is “correctly approved” this means that the models predicted that the client would pay back the loan successfully and they were correct. However, this also represents forgone revenue because this is median amount that Homecredit makes from a loan. Consequently this means that if they reject someone who could have paid back a loan, they lose the potential to make 23,800.5 dollars from that client.

“Incorrect Loan Approval” is when a client is approved for a loan, but then they are unable to pay it off. This results in a median loss of 24,412.5 dollars for Homecredit.

### Solution:

```
# Store all the values derived from the previous analysis into variables
XGB_incorrect_approval <- 24363.0

XGB_incorrect_count <- 17940

MCC_incorrect_approval <- 24412.5

MCC_incorrect_count <- 19245

# Use the variables to calculate the incorrect_approval_difference values
incorrect_approval_diff <- MCC_incorrect_approval - XGB_incorrect_approval

# Calculate the scaled median loss per false positive
median_loss_per_false_positive_scaled_diff <- (MCC_incorrect_count * incorrect_approval_diff) - (XGB_incorrect_count * MCC_incorrect_approval)

median_loss_per_false_positive_scaled_diff # Display the results
```

[1] 64597.5

This is the scaled aggregate savings from using the XGBoost model compared to the majority classifier, purely based on the \$49 per-instance median loss difference across all incorrect loan approvals. The value also represents the overall savings that come purely from the 49 dollar median loss reduction across all the incorrect loan approvals regardless of the actual counts.

This also means that the Xgboost model is doing a good job since the count of incorrect approvals is much lower when compared to the Majority Classifier Model. (17940 - XGBoost vs 19245 - Majority Classifier). Furthermore, the median incorrect loan approval values are lower, which means that XGboost's mistakes are less harmful when compared to predicting with the majority. (24,412 - Majority Classifier vs 24,363 - XGboost)

XGBoost however did not decrease the loss of forgone revenue (23800.5 for both XGboost and Majority Classifier), but it did not increase it either. The value of forgone revenue, 23,800 dollars remained the same across both models.