Haolin Luo DeFi F24 Final Notebook

Haolin Luo

15 December 2024

0.1 0.0 Preliminaries.

0.2 DAR Project and Group Members

- Project name: DeFi Claasification group
- Project team members: Haolin Luo, Hanzhen Qin, Zihan Nie

This report is generated from an R Markdown file that includes all the R code necessary to produce the results described and embedded in the report. Code blocks can be surpressed from output for readability using the command code `{R, echo=show}` in the code block header. If `show <- FALSE` the code block will be surpressed; if `show <- TRUE` then the code will be show.

Executing this R notebook requires some subset of the following packages:

- `ggplot2`
- 'rmarkdown' *'tidyverse'* 'stringr' *'ggbiplot'* 'pheatmap' *'randomForest'* 'caret' *'survival'* 'survminer' *'ggplot2'* 'kableExtra' *'rpart'

These will be installed and loaded as necessary (code suppressed).

1 1.0 Project Introduction

Overall, all of my code is related to classification models, also some code for smote results with in terms of some imbalanced datasets, mainly the event pairs with liquidations.

For data processing, we get the datasets from our survival data folder based on different index event and outcome event. For index events, we have Borrow/Deposit/Repay/Withdraw. Besides, for outcome events, we have Account Liquidated/Liquidation Performed/Deposit/Repay/Withdraw/Borrow. We consider each different index event and outcome event as a pair, such as “Borrow” and “Repay”, “Deposit” and “Borrow”. For our classification, we need to apply survival analysis and turn the given datasets into a binary problem based on the time threshold. After the process, the binary event column will be “event” with “yes” or “no”. In order to make the dataset events more balanced, we use cutoff days as time threshold based on RMST calculations. In general, we will have 16 different pairs in total. (We drop all the event pairs that contain “liquidation performed”) We run the all the models we have based on the datasets after processing. Our model includes Logistic Regression, K nearest neighbor, Random forest, Decision Tree, Naive Bayes, XG boost, ADA boost, GBM, Elastic Net. We compared the results of each model and try to conclude which model is the best.

We checked the percent of event for each event pair, we found that some datasets are unbalanced. To fix that problem, I use the SMOTE function in our pipeline to balance our train dataset and use the new dataset to train the model. Then I will test trained model on test dataset to check if we get better balanced accuracy.

1.1 Summary of github files

luoh4_Merge_Module.Rmd: It contains the code for all the model function that we are going to use for the final pipeline. Models include Naive Bayes, XG boost, ADA boost, GBM, Elastic Net. luoh4_Assignment-5-models.Rmd: It contains the current models that we run on our dataset with different event pairs with their corresponding cutoff days. I used for loop for each model to make sure it will run through every different event pairs but Liquidatation Performed. luoh4_Assignment5_analysis.Rmd: It contains the result of all models' result on our event pairs, it is the result from luoh4_Assignment-5-models. And I draw some plots to make the result more clearly. luoh4_final_notebook.Rmd: The final notebook drafts that contain all the results we find with the current dataset and all my personal contribution to the classification group. Pipeline_new_data.Rmd： The main code file that has different models run on our new dataset. The SMOTE is also included in this code file. Final_Check_percent_of_events.Rmd: It uses the function "get_percentage" in pipeline to check the event percentage in each train dataset.

- Summary of github commits
    - branch name: dar-luoh4
    - filenames: luoh4_Merge_Module.Rmd, luoh4_Assignment-5-models.Rmd, luoh4_Assignment5_analysis.Rmd, luoh4_final_notebook.Rmd, Pipeline_new_data.Rmd, Final_Check_percent_of_events.Rmd

1.2 Personaly Contribution

1. Running all the models on original datasets and produce visual graphs for those models' results based on their balanced accuracy.
2. Create the function of five models, including Naive Bayes, XG boost, ADA boost, GBM, Elastic Net
3. Check the percentage of event in each event pairs and identify the unbalanced dataset whose percentage is greater than 90%.
4. Using SMOTE function to balanced some unbalanced datasets and get the balanced accuracy after SMOTE.

2 2.0 Organization of Report

This report is organize as follows:

- Section 3.1. Finding 1: Name: Model performance on dataset Running all current models throught the given dataset. Using for loop to go over each event pairs with their corresponding RMST cutoffs, and draw some graphs to find the best model based on balanced accuracy.
- Section 4.0. Finding 2: Name: Merge functions Description: Write my models into function so that it can be merged into pipeline. Each function will take train and test data as input and return a confusion matrix.
- Section 5.0. Finding 3: Name: Use SMOTE to deal with the unbalanced datasets. Description: Some event pairs' data are unbalaced and our models show a bad performance on those datasets. In order to fix the problem, I use SMOTE function in pipeline to balanced the train dataset so that we can train our model and use the trained model to predict the test data. Then we will compare the result before SMOTE and after SMOTE to check if our SMOTE function is working.
- Section 6.0. Finding 4: Name: Running models on new datasets Description: Running different models on the dataset after feature creation. Models included: XG boost, Ada boost, Naive Bayes, GBM.
- Section 7.0. Finding 5: Name: Check event percentage across different event pairs Description: To find the unbalanced datasets, we go through each event pairs and identify the event pairs that percentage of event (proportion of 'yes' V.S. 'no') is greater than 90% as unbalanced.
- Section 8.0 Overall conclusions and suggestions
- Section 9.0 Appendix: This section describe the following additional works that may be helpful in the future work:

3 3.0 Finding 1: Model performance on dataset

Question: We are tring to find a best model on the given dataset based on their final balanced accuracy.

Approach: Using for loop to run each model on different event pairs with their corresponding RMST cutoffs. The dataset are pre-processed as I stated in Introduction.

Result: We get the model performance with balanced accuracy as our output and save them into rds files. Then we draw graph to analyze the results and find the best model.

3.1 3.1 Data, Code, and Resources

Here is a list data sets, codes, that are used in your work. Along with brief description and URL where they are located.

1. luoh4-Assignment-5-citpff-percent.Rmd, the code for running models on different event pairs with RMST cutoff [https://github.rpi.edu/DataINCITE/DAR-DeFi-LTM-F24/blob/main/StudentNotebooks/Assignment03/luoh4-Assignment-5-cutoff-percent.Rmd (https://github.rpi.edu/DataINCITE/DAR-DeFi-LTM-F24/blob/main/StudentNotebooks/Assignment03/luoh4-Assignment-5-cutoff-percent.Rmd)]

2. confusion_matrix_results_RMST_5_EN.rds [https://github.rpi.edu/DataINCITE/DAR-DeFi-LTM-F24/blob/main/StudentNotebooks/Assignment03/confusion_matrix_results_RMST_5_EN.rds (https://github.rpi.edu/DataINCITE/DAR-DeFi-LTM-F24/blob/main/StudentNotebooks/Assignment03/confusion_matrix_results_RMST_5_EN.rds)] confusion_matrix_results_RMST_5_GBM.rds [https://github.rpi.edu/DataINCITE/DAR-DeFi-LTM-F24/blob/main/StudentNotebooks/Assignment03/confusion_matrix_results_RMST_5_GBM.rds (https://github.rpi.edu/DataINCITE/DAR-DeFi-LTM-F24/blob/main/StudentNotebooks/Assignment03/confusion_matrix_results_RMST_5_GBM.rds)] confusion_matrix_results_RMST_5_XG.rds [https://github.rpi.edu/DataINCITE/DAR-DeFi-LTM-F24/blob/main/StudentNotebooks/Assignment03/confusion_matrix_results_RMST_5_XG.rds (https://github.rpi.edu/DataINCITE/DAR-DeFi-LTM-F24/blob/main/StudentNotebooks/Assignment03/confusion_matrix_results_RMST_5_XG.rds)] confusion_matrix_results_RMST_5_ada.rds [https://github.rpi.edu/DataINCITE/DAR-DeFi-LTM-F24/blob/main/StudentNotebooks/Assignment03/confusion_matrix_results_RMST_5_ada.rds (https://github.rpi.edu/DataINCITE/DAR-DeFi-LTM-F24/blob/main/StudentNotebooks/Assignment03/confusion_matrix_results_RMST_5_ada.rds)] confusion_matrix_results_RMST_5_dt.rds [https://github.rpi.edu/DataINCITE/DAR-DeFi-LTM-F24/blob/main/StudentNotebooks/Assignment03/confusion_matrix_results_RMST_5_dt.rds (https://github.rpi.edu/DataINCITE/DAR-DeFi-LTM-F24/blob/main/StudentNotebooks/Assignment03/confusion_matrix_results_RMST_5_dt.rds)] confusion_matrix_results_RMST_5_knn.rds [https://github.rpi.edu/DataINCITE/DAR-DeFi-LTM-F24/blob/main/StudentNotebooks/Assignment03/confusion_matrix_results_RMST_5_knn.rds (https://github.rpi.edu/DataINCITE/DAR-DeFi-LTM-F24/blob/main/StudentNotebooks/Assignment03/confusion_matrix_results_RMST_5_knn.rds)] confusion_matrix_results_RMST_5_lr.rds [https://github.rpi.edu/DataINCITE/DAR-DeFi-LTM-F24/blob/main/StudentNotebooks/Assignment03/confusion_matrix_results_RMST_5_lr.rds (https://github.rpi.edu/DataINCITE/DAR-DeFi-LTM-F24/blob/main/StudentNotebooks/Assignment03/confusion_matrix_results_RMST_5_lr.rds)] confusion_matrix_results_RMST_5_nb.rds [https://github.rpi.edu/DataINCITE/DAR-DeFi-LTM-F24/blob/main/StudentNotebooks/Assignment03/confusion_matrix_results_RMST_5_nb.rds (https://github.rpi.edu/DataINCITE/DAR-DeFi-LTM-F24/blob/main/StudentNotebooks/Assignment03/confusion_matrix_results_RMST_5_nb.rds)] confusion_matrix_results_combined.rds [https://github.rpi.edu/DataINCITE/DAR-DeFi-LTM-F24/blob/main/StudentNotebooks/Assignment03/confusion_matrix_results_combined.rds (https://github.rpi.edu/DataINCITE/DAR-DeFi-LTM-F24/blob/main/StudentNotebooks/Assignment03/confusion_matrix_results_combined.rds)] Those rds are the result from “luoh4-Assignment-5-cutpff-percent.Rmd”

3. luoh4_Assignment5_analysis.Rmd, the code for genrating graph and plot for models’ performance on given dataset with RMST cutoffs. [https://github.rpi.edu/DataINCITE/DAR-DeFi-LTM-F24/blob/main/StudentNotebooks/Assignment03/luoh4_Assignment5_analysis.Rmd (https://github.rpi.edu/DataINCITE/DAR-DeFi-LTM-F24/blob/main/StudentNotebooks/Assignment03/luoh4_Assignment5_analysis.Rmd)]

Creating the data frame of each event pair with their corresponding RMST cutoff. Then using a for loop to do the data processing and feature dropping for each data and run model on them. The full code can be found in “luoh4-Assignment-5-citpff-percent.Rmd”

3.2 3.2 Contribution

Solo work

3.3 3.3 Methods Description

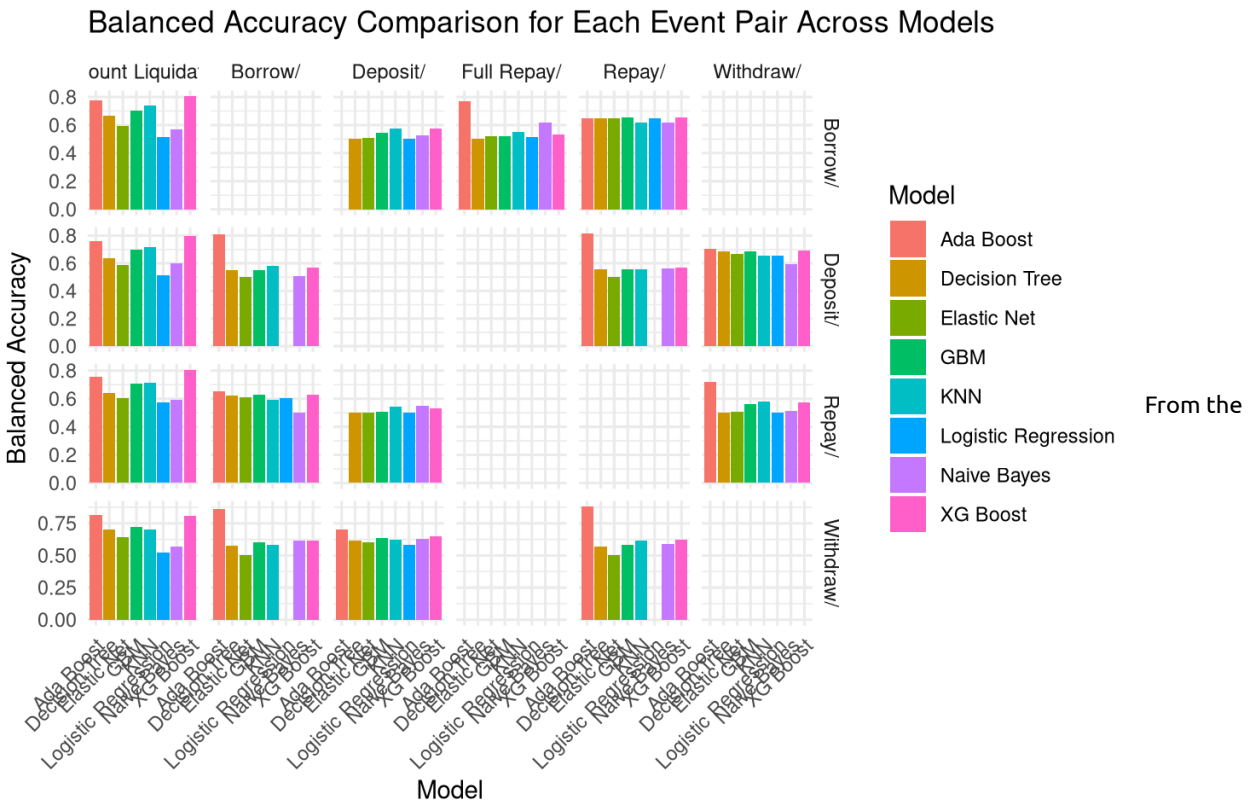
No specific methods are used for this part. Mainly running models on different dataset after data processing. Models include: Logistic Regression, K nearest neighbor, Random forest, Decision Tree, Naive Bayes, XG boost, ADA boost, GBM, Elastic Net.

3.4 3.4 Result and Discussion

```
combined_results <- readRDS("~/DAR-DeFi-LTM-F24/StudentNotebooks/Assignment03/confusion_matrix_results_combined.rds")
print(colnames(combined_results))
```

```
## [1] "IndexEvent"       "OutcomeEvent"     "Accuracy"         "Sensitivity"
## [5] "Specificity"      "BalancedAccuracy" "Model"
```

```
# Filter and organize data for the plot
plot_data <- combined_results %>%
  dplyr::select("IndexEvent", "OutcomeEvent", "Model", "BalancedAccuracy") %>%
  drop_na()


# Create the bar plot
ggplot(plot_data, aes(x = Model, y = BalancedAccuracy, fill = Model)) +
  geom_bar(stat = "identity", position = "dodge") +
  facet_grid(IndexEvent ~ OutcomeEvent, scales = "free_y") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(
    title = "Balanced Accuracy Comparison for Each Event Pair Across Models",
    x = "Model",
    y = "Balanced Accuracy"
  )
```



From the graph, we can find that in general, the boosing methods are performing better than other methods. Ada boost outperformed XG boost on some datasets.

4 4.0 Finding 2: Create function for models

Question: Merge all models we have into our pipeline.

Approach: Write each model into different functions. Take train data and test data as input. Once we use the function, it will train the model and get the prediction result. Finally, the function will return a confusion matrix. Models included: Elastic Net, XG boost, GBM, SVM, Ada boost, Naive Bayes

Result: Get the confusion matrix as a return.

4.1 4.1 Code and Description

Full Code can be found as “luoh4_Merge_Module.Rmd”. Link: [https://github.rpi.edu/DataINCITE/DAR-DeFi-LTM-F24/blob/main/StudentNotebooks/Assignment05/luoh4_Merge_Module.Rmd (https://github.rpi.edu/DataINCITE/DAR-DeFi-LTM-F24/blob/main/StudentNotebooks/Assignment05/luoh4_Merge_Module.Rmd)]

Elastic_net function: The get_elastic_net function is designed to train and evaluate an Elastic Net regression model for binary classification tasks. It takes a training dataset and a test dataset as inputs, splits the data into features and the target variable, trains the model using cross-validation, and evaluates its performance through a confusion matrix.

XG boost function: The run_xgboost function is designed for training and evaluating an XGBoost model in binary classification tasks. It allows flexible input of training and testing datasets, along with customizable model hyperparameters. By default, it sets parameters such as a logistic objective function, error as the evaluation metric, and specific tree and learning settings for efficient boosting. The function processes data by converting categorical variables to numeric and prepares feature matrices for compatibility with XGBoost. It trains the model using the specified parameters and number of iterations, then predicts class probabilities on the test set, converting these into binary labels. A confusion matrix is generated to evaluate model performance, which is returned and printed. The function ensures reproducibility through a fixed random seed, making it a robust and versatile tool for binary classification modeling.

GBM function: The run_gbm function is a tool for training and evaluating a Gradient Boosting Machine (GBM) model for binary classification. It accepts training and testing datasets, specifies the target variable, and includes hyperparameters such as the number of trees, tree depth, learning rate, and minimum observations per node. The function processes the data by converting the target variable into numeric format for compatibility with the GBM framework and prepares the feature matrix for prediction. It trains the GBM model using a “bernoulli” distribution, optimizing performance through cross-validation to determine the best number of boosting iterations. Predictions are made on the test dataset,

converting probabilities into binary class labels. A confusion matrix is generated to evaluate the model's predictive performance, which is returned alongside the identified optimal number of trees, ensuring efficient and interpretable model evaluation.

SVM function: The run_svm function is designed for training and evaluating a Support Vector Machine (SVM) model for binary classification. It accepts training and testing datasets, the target column name, and the kernel type as input. The function ensures the target variable in the training data is converted to a factor and prepares the testing dataset by separating features from the target variable. It trains the SVM model using the specified kernel type (defaulting to "linear") and makes predictions on the test set. Predictions and actual labels are harmonized to ensure consistency in their levels. The function evaluates the model's performance by generating a confusion matrix, which highlights the prediction accuracy and misclassification rates. The confusion matrix is returned as the output, providing a concise summary of the model's classification performance.

Ada boost function: The run_ada function is an implementation for training and evaluating an Ada boost model on a binary classification task. It takes as input the training and testing datasets, the target column, and the number of boosting iterations (n_iter, defaulting to 50). The function ensures that the target column in the training data is converted to a factor. The test dataset is prepared by separating features from the target variable. The AdaBoost model is then trained using the specified number of boosting iterations. Predictions are generated for the test dataset, and a confusion matrix is created to evaluate the model's performance by comparing predicted labels against actual target labels. The confusion matrix, summarizing the model's predictive accuracy and errors, is returned as the output.

Naive_Bayes function: The run_naive_bayes function implements a Naive Bayes classifier for binary classification tasks. It takes in training and testing datasets, along with the target column, and ensures the target variable in both datasets is treated as a factor. The test dataset is prepared by excluding the target column for prediction purposes. Using the training data, the Naive Bayes model is trained on all other features. Predictions are made on the test data, and the predicted labels are adjusted to match the levels of the actual test labels. The function evaluates the model's performance by constructing a confusion matrix that compares the predicted and actual labels. The resulting confusion matrix, which summarizes the classification results, is returned as the output.

4.2 4.2 Contribution

Solo work

5 5.0 Finding 3 Use SMOTE to deal with the unbalanced datasets.:

Question: Since some datasets are unbalanced so that I try to use the SMOTE function in our pipeline to balance our train dataset and test the new model on test dataset which is still unbalanced.

Approach: After split the liquidation dataset, I will oversample the trainset and train model based on new trainset. After getting the model from trainning, I will use it to predict the test dataset.

Result: We found that before SMOTE, Ada boost and GBM are having the best performance. Whereas, the Naive Bayes becomes the best model after SMOTE.

5.1 5.1 Data, Code, and Resources

1. Pipeline_new_data.Rmd (line 678-1173) [https://github.rpi.edu/DataINCITE/DAR-DeFi-LTM-F24/blob/main/StudentNotebooks/Assignment07_DraftFinalProjectNotebook/Pipeline_new_data.Rmd (https://github.rpi.edu/DataINCITE/DAR-DeFi-LTM-F24/blob/main/StudentNotebooks/Assignment07_DraftFinalProjectNotebook/Pipeline_new_data.Rmd)]
2. confusion_matrix_results_unbalanced.rds, includes the final result after SMOTE [https://github.rpi.edu/DataINCITE/DAR-DeFi-LTM-F24/blob/main/StudentNotebooks/Assignment07_DraftFinalProjectNotebook/confusion_matrix_results_unbalanced.rds (https://github.rpi.edu/DataINCITE/DAR-DeFi-LTM-F24/blob/main/StudentNotebooks/Assignment07_DraftFinalProjectNotebook/confusion_matrix_results_unbalanced.rds)]

5.2 5.2 Contribution

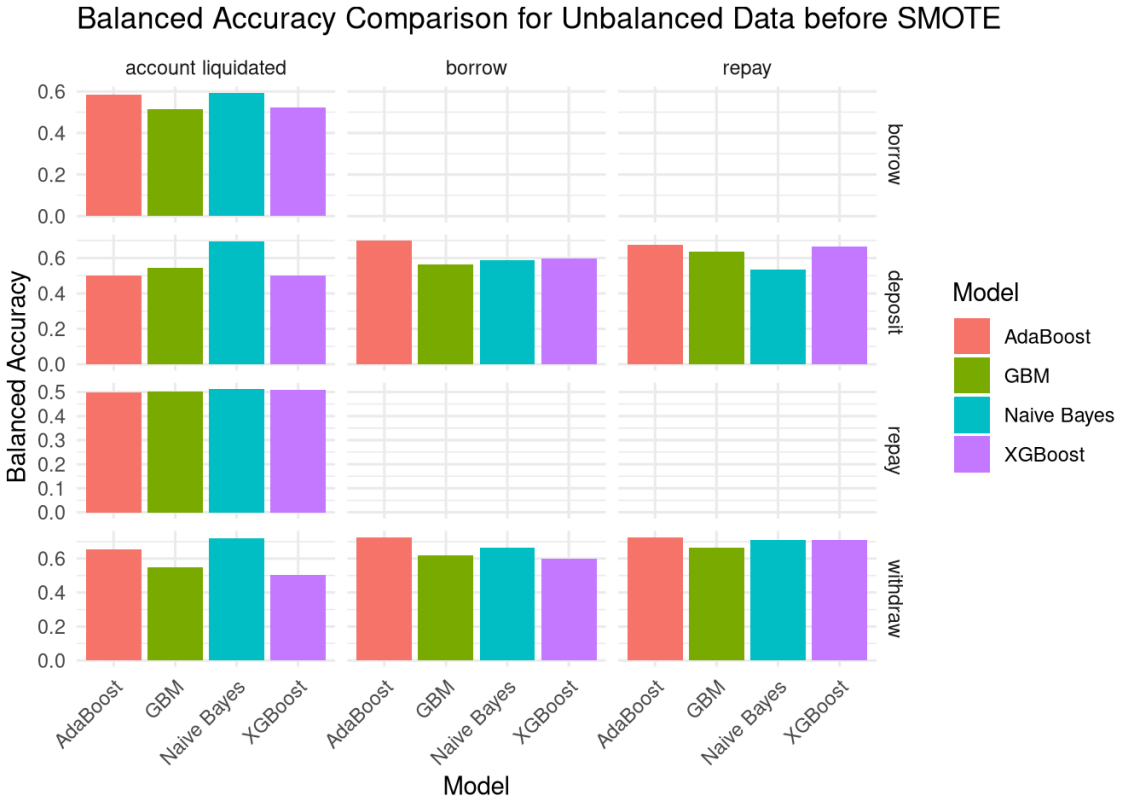
Use the SMOTE function in our pipeline that created by Hanzhen. The rest of part is done by myself alone.

5.3 5.3 Methods Description

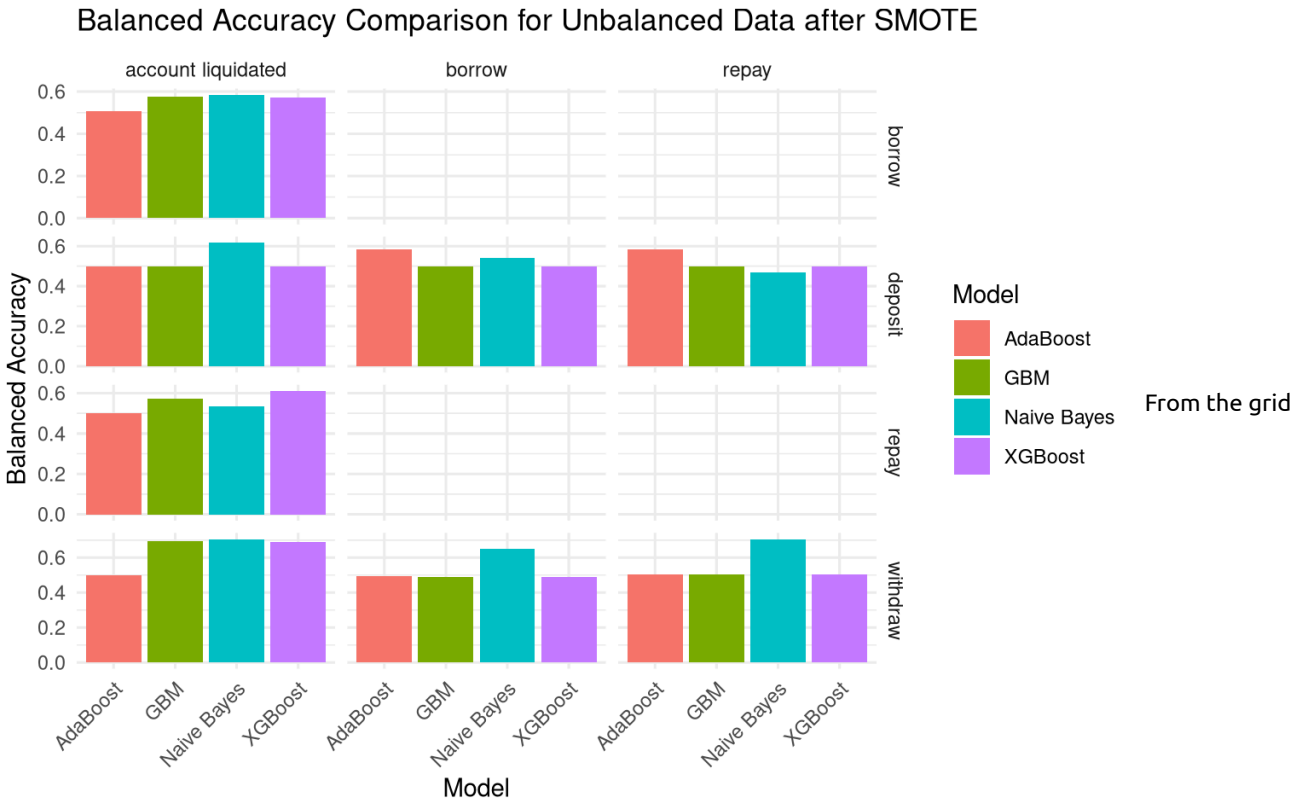
After grab the data with our pipeline function, use the SMOTE function in pipeline to make our train dataset more balanced. Then we will train the model based on the new train dataset and use our test dateset to check if the balanced accuracy increase.

5.4 5.4 Result and Discussion

```
df_6 <- data.frame(
  IndexEvent = c("borrow",
                 "deposit", "deposit",  "deposit",
                 "repay",
                 "withdraw", "withdraw", "withdraw"),
  OutcomeEvent = c("account liquidated",
                   "account liquidated", "borrow",  "repay",
                   "account liquidated",
                   "account liquidated", "borrow", "repay")
)
combined_results <- readRDS("~/DAR-DeFi-LTM-F24/StudentNotebooks/Assignment07_DraftFinalProjectNotebook/confusion_matrix_results.rds")
combined_results_unbalanced <- readRDS("~/DAR-DeFi-LTM-F24/StudentNotebooks/Assignment07_DraftFinalProjectNotebook/confusion_matrix_results_unbalanced.rds")

# Filter combined_results to include only event pairs in df_6
filtered_combined_results <- combined_results %>%
  inner_join(df_6, by = c("IndexEvent", "OutcomeEvent"))

ggplot(filtered_combined_results, aes(x = Model, y = BalancedAccuracy, fill = Model)) +
  geom_bar(stat = "identity", position = "dodge") +
  facet_grid(IndexEvent ~ OutcomeEvent, scales = "free_y") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(
    title = "Balanced Accuracy Comparison for Unbalanced Data before SMOTE",
    x = "Model",
    y = "Balanced Accuracy"
  )
```



```
# Filter and organize data for the plot
plot_data_unbalanced <- combined_results_unbalanced %>%
  dplyr::select(IndexEvent, OutcomeEvent, Model, BalancedAccuracy)
plot_data_unbalanced$BalancedAccuracy <- as.numeric(plot_data_unbalanced$BalancedAccuracy)

# Load ggplot2 library
library(ggplot2)

# Assuming 'plot_data_unbalanced' is your dataset
# Create the bar plot
ggplot(plot_data_unbalanced, aes(x = Model, y = BalancedAccuracy, fill = Model)) +
  geom_bar(stat = "identity", position = "dodge") +
  facet_grid(IndexEvent ~ OutcomeEvent, scales = "free_y") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(
    title = "Balanced Accuracy Comparison for Unbalanced Data after SMOTE",
    x = "Model",
    y = "Balanced Accuracy"
  )
```



plot, we can found that our Naive bayes model has the best performance among all the event pairs after SMOTE. At the same time, the accuracy of XG boost and Ada boost have a huge decrease on their balanced accuracy after SMOTE.

6 6.0 Finding 4 Running current models on new dataset:

Question: Which is the best model on our newest dataset?

Approach: Use the data processing code from our pipeline to get get train and test dataset. Then train the model based on the train dataset.

Result: From the result, we can conclude that Ada boost and GBM boost are performing well in general. Although for unbalanced datasets they have a bad performance, overall these two models are the best.

6.1 6.1 Data, Code, and Resources

Pipeline_new_data.Rmd (line 174-677), in this part, it has all the code of runnning models on different event pairs. Models include Elastic Net, SVM, Ada boost, XG boost, GBM and Naive Bayes [https://github.rpi.edu/DataINCITE/DAR-DeFi-LTM-F24/blob/main/StudentNotebooks/Assignment07_DraftFinalProjectNotebook/Pipeline_new_data.Rmd (https://github.rpi.edu/DataINCITE/DAR-DeFi-LTM-F24/blob/main/StudentNotebooks/Assignment07_DraftFinalProjectNotebook/Pipeline_new_data.Rmd)]

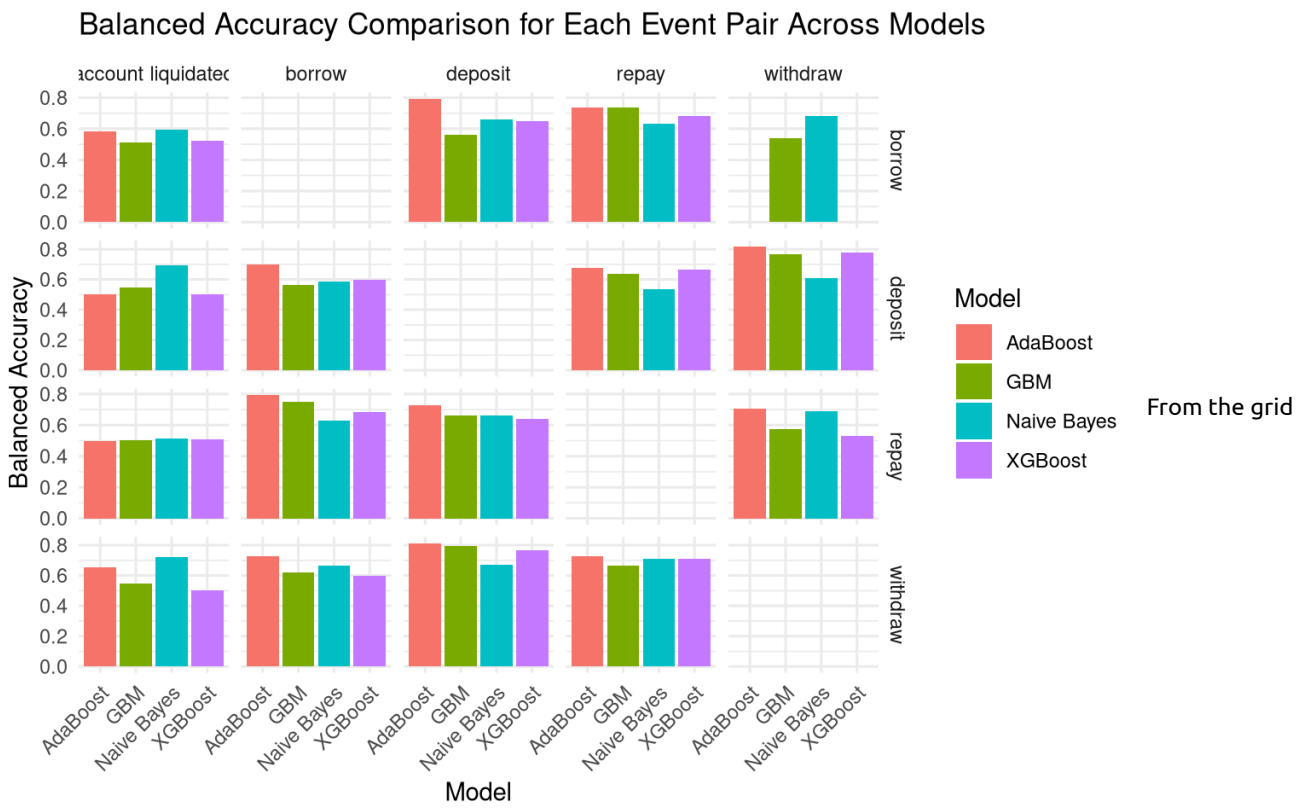
confusion_matrix_results.rds, it contains the result of different models' accuracy on all event pairs except "liquidation performed" [https://github.rpi.edu/DataINCITE/DAR-DeFi-LTM-F24/blob/main/StudentNotebooks/Assignment07_DraftFinalProjectNotebook/confusion_matrix_results.rds (https://github.rpi.edu/DataINCITE/DAR-DeFi-LTM-F24/blob/main/StudentNotebooks/Assignment07_DraftFinalProjectNotebook/confusion_matrix_results.rds)] ## 6.2 Contribution Use the data processing functions that created by Hanzhen, and I use my model to do the classification test.

6.2 6.3 Methods Description

For data processing part, I direct use the pipeline, and get both train and test datasets. Then I will use default parameter for each model to train the model then use the model to test the balanced accuracy based on test dataset.

6.3 6.4 Result and Discussion

```
combined_results <- readRDS("~/DAR-DeFi-LTM-F24/StudentNotebooks/Assignment07_DraftFinalProjectNotebook/confusion_matrix_results.rds")

# Filter and organize data for the plot
plot_data <- combined_results %>%
  dplyr::select(IndexEvent, OutcomeEvent, Model, BalancedAccuracy)

# Create the bar plot
ggplot(plot_data, aes(x = Model, y = BalancedAccuracy, fill = Model)) +
  geom_bar(stat = "identity", position = "dodge") +
  facet_grid(IndexEvent ~ OutcomeEvent, scales = "free_y") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(
    title = "Balanced Accuracy Comparison for Each Event Pair Across Models",
    x = "Model",
    y = "Balanced Accuracy"
  )
```



plot, we can see that Ada boost and GBM outperformed other models in general. However, they are not good at unbalanced datasets which will be introduced on next part on my notebook.

7 7.0 Finding 5 Check event percentage across different event pairs:

Question: Which dataset is unbalanced?

Approach: Set the threshold to be 90% which means if the event percentage (“yes” or “no”) in our final result is greater than 90%, we will consider it as unbalanced data and use SMOTE to balance the corresponding train dataset.

Result: We have 8 unbalanced event pairs in total, including borrow, account liquidated deposit, account liquidated deposit, borrow deposit, repay repay, account liquidated withdraw, account liquidated withdraw, borrow withdraw, repay

All the listed event pairs have unbalanced datasets.

7.1 7.1 Data, Code, and Resources

Final_Check_percent_of_events.Rmd, it uses the function “get_percentage” in pipeline to check the event percentage in each train dataset. [https://github.rpi.edu/DataINCITE/DAR-DeFi-LTM-F24/blob/main/StudentNotebooks/Assignment07_DraftFinalProjectNotebook/Final_Check_percent_of_events.Rmd (https://github.rpi.edu/DataINCITE/DAR-DeFi-LTM-F24/blob/main/StudentNotebooks/Assignment07_DraftFinalProjectNotebook/Final_Check_percent_of_events.Rmd)]

7.2 7.2 Contribution

Use the “get_percentage” function in pipeline which is created by Hanzhen

7.3 7.3 Methods Description

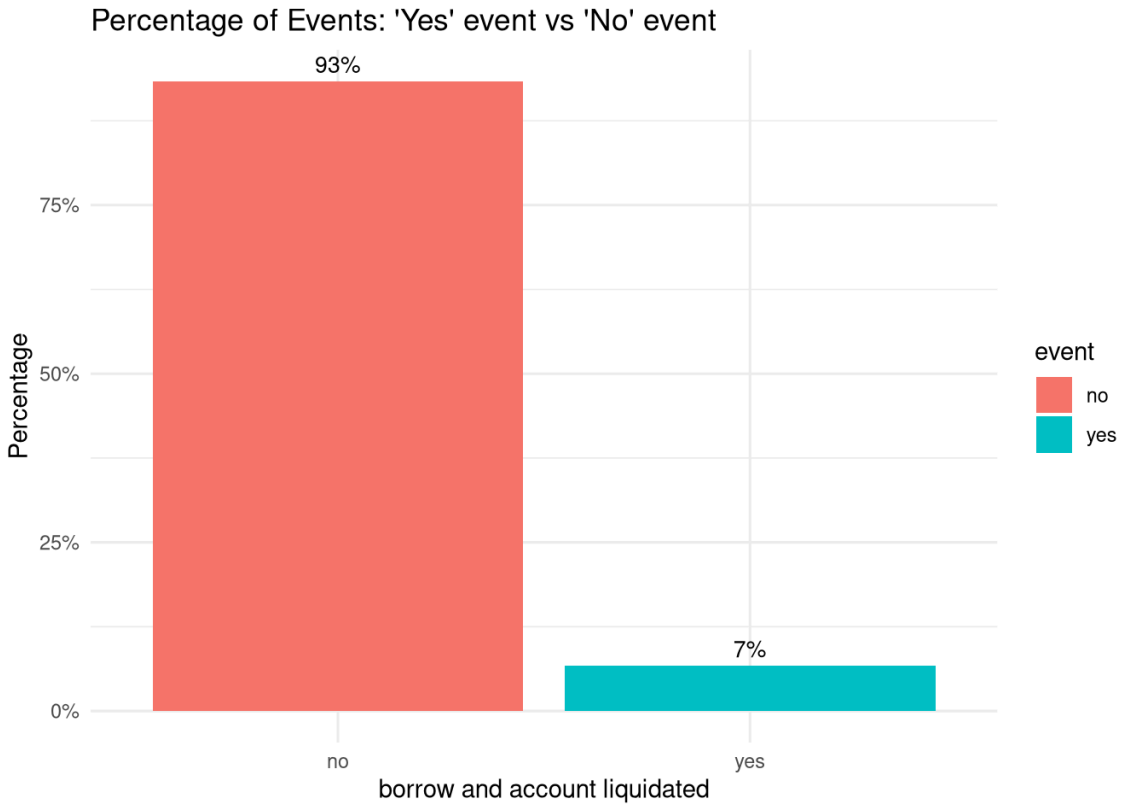
Set the threshold to be 90% which means if the event percentage (“yes” or “no”) in our final result is greater than 90%, we will consider it as unbalanced data and use SMOTE to balance the corresponding train dataset.

7.4 7.4 Result and Discussion

```
for (i in 1:nrow(df_6)) {
  indexEvent <- df_6$IndexEvent[i]
  outcomeEvent <- df_6$OutcomeEvent[i]

  # load the corresponding train and test data
  get_train_test_data(indexEvent, outcomeEvent)


  classification_cutoff = get_classification_cutoff(indexEvent, outcomeEvent)
  train_data = data_processing(train, classification_cutoff)
  get_percentage(train_data, indexEvent, outcomeEvent)
}
```

```
## Warning in inner_join(y, X, by = "id"): Detected an unexpected many-to-many relationship between `x` and
`y`.
## ℹ Row 570 of `x` matches multiple rows in `y`.
## ℹ Row 637 of `y` matches multiple rows in `x`.
## ℹ If a many-to-many relationship is expected, set `relationship =
##   "many-to-many"` to silence this warning.
```

```
## Warning in inner_join(y, X, by = "id"): Detected an unexpected many-to-many relationship between `x` and
`y`.
## ℹ Row 82 of `x` matches multiple rows in `y`.
## ℹ Row 82 of `y` matches multiple rows in `x`.
## ℹ If a many-to-many relationship is expected, set `relationship =
##   "many-to-many"` to silence this warning.
```

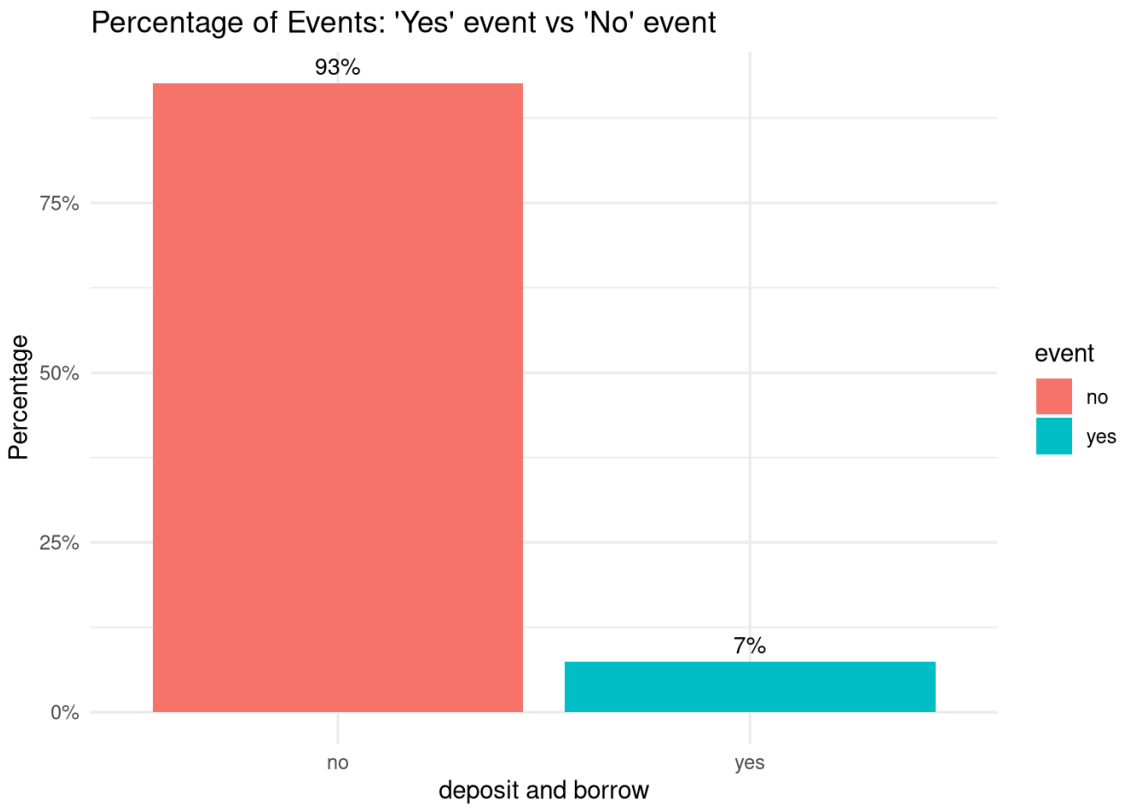



```
## Warning in inner_join(y, X, by = "id"): Detected an unexpected many-to-many relationship between `x` and
`y`.
## ℹ Row 62 of `x` matches multiple rows in `y`.
## ℹ Row 62 of `y` matches multiple rows in `x`.
## ℹ If a many-to-many relationship is expected, set `relationship =
##   "many-to-many"` to silence this warning.
```

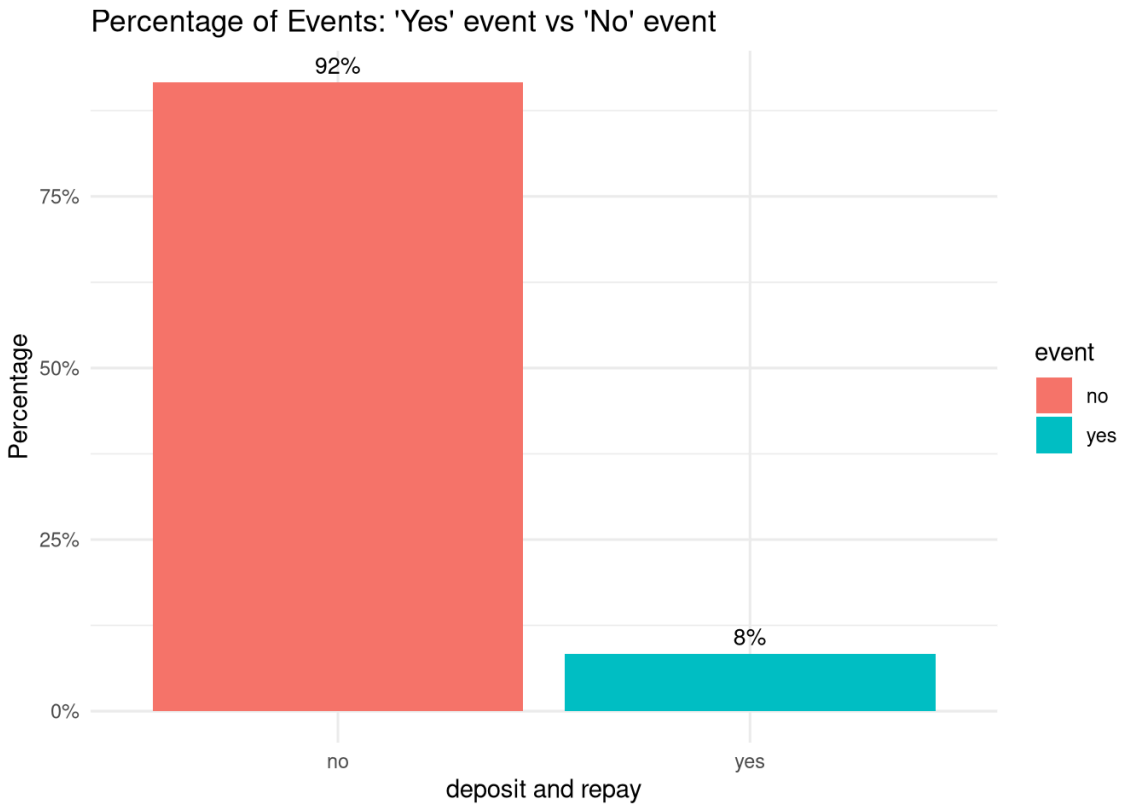
```
## Warning in inner_join(y, X, by = "id"): Detected an unexpected many-to-many relationship between `x` and
`y`.
## ℹ Row 44 of `x` matches multiple rows in `y`.
## ℹ Row 44 of `y` matches multiple rows in `x`.
## ℹ If a many-to-many relationship is expected, set `relationship =
##   "many-to-many"` to silence this warning.
```


```
## Warning in inner_join(y, X, by = "id"): Detected an unexpected many-to-many relationship between `x` and
`y`.
## ℹ Row 1365 of `x` matches multiple rows in `y`.
## ℹ Row 1612 of `y` matches multiple rows in `x`.
## ℹ If a many-to-many relationship is expected, set `relationship =
##   "many-to-many"` to silence this warning.
```

```
## Warning in inner_join(y, X, by = "id"): Detected an unexpected many-to-many relationship between `x` and
`y`.
## ℹ Row 33 of `x` matches multiple rows in `y`.
## ℹ Row 44 of `y` matches multiple rows in `x`.
## ℹ If a many-to-many relationship is expected, set `relationship =
##   "many-to-many"` to silence this warning.
```

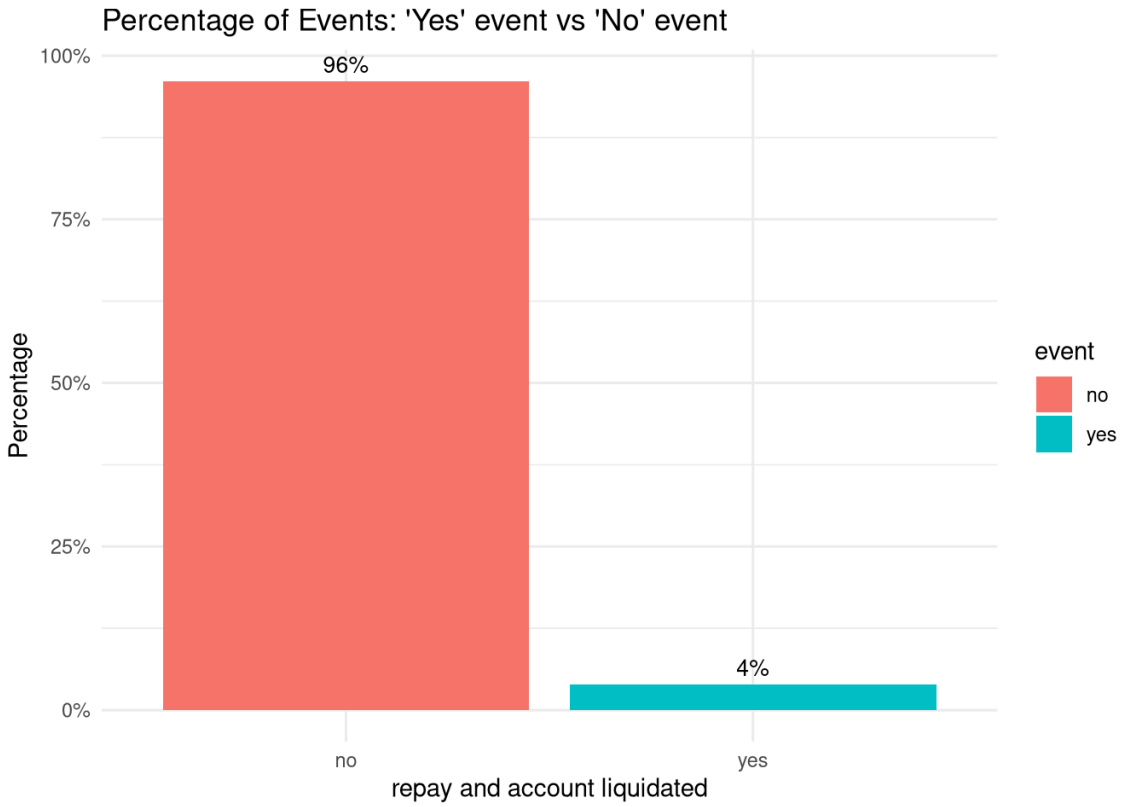



```
## Warning in inner_join(y, X, by = "id"): Detected an unexpected many-to-many relationship between `x` and
`y`.
## Detected an unexpected many-to-many relationship between `x` and `y`.
## ℹ Row 25 of `x` matches multiple rows in `y`.
## ℹ Row 62 of `y` matches multiple rows in `x`.
## ℹ If a many-to-many relationship is expected, set `relationship =
##   "many-to-many"` to silence this warning.
```



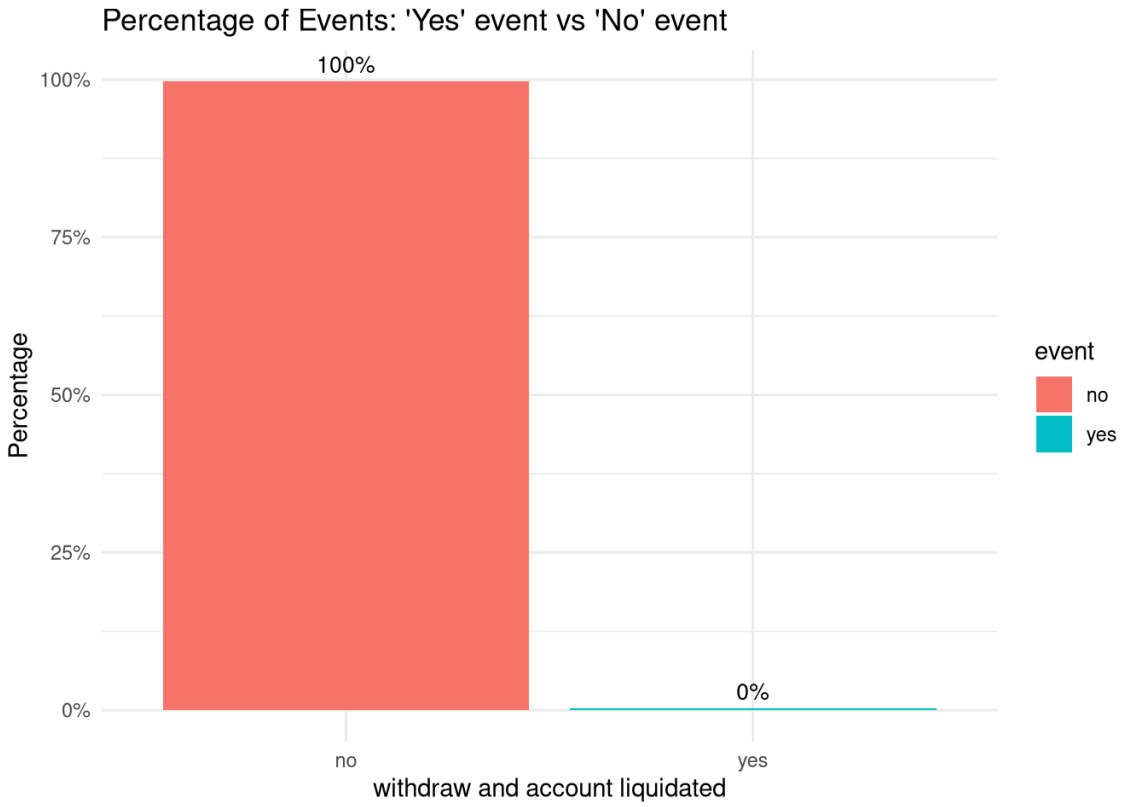
```
## Warning in inner_join(y, X, by = "id"): Detected an unexpected many-to-many relationship between `x` and
`y`.
## ℹ Row 7449 of `x` matches multiple rows in `y`.
## ℹ Row 8230 of `y` matches multiple rows in `x`.
## ℹ If a many-to-many relationship is expected, set `relationship =
##   "many-to-many"` to silence this warning.
```

```
## Warning in inner_join(y, X, by = "id"): Detected an unexpected many-to-many relationship between `x` and
`y`.
## ℹ Row 1777 of `x` matches multiple rows in `y`.
## ℹ Row 1809 of `y` matches multiple rows in `x`.
## ℹ If a many-to-many relationship is expected, set `relationship =
##   "many-to-many"` to silence this warning.
```



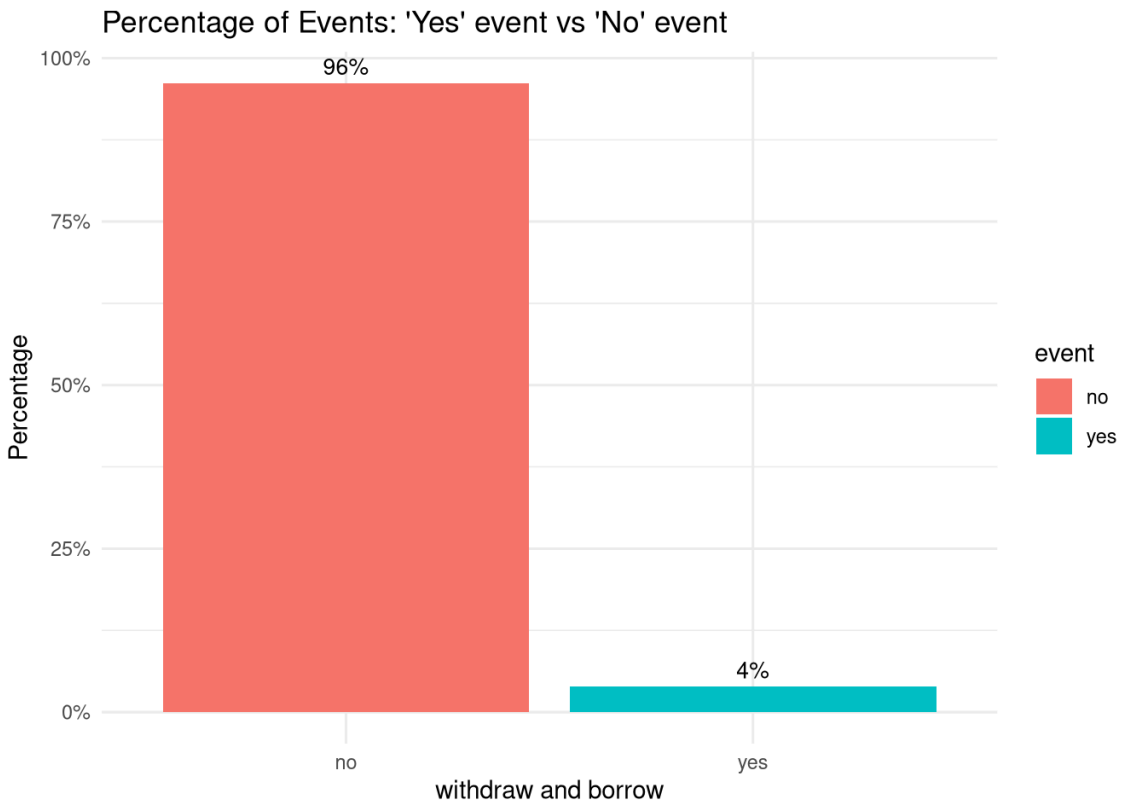

```
## Warning in inner_join(y, X, by = "id"): Detected an unexpected many-to-many relationship between `x` and
`y`.
## ℹ Row 29 of `x` matches multiple rows in `y`.
## ℹ Row 29 of `y` matches multiple rows in `x`.
## ℹ If a many-to-many relationship is expected, set `relationship =
##   "many-to-many"` to silence this warning.
```

```
## Warning in inner_join(y, X, by = "id"): Detected an unexpected many-to-many relationship between `x` and
`y`.
## ℹ Row 33 of `x` matches multiple rows in `y`.
## ℹ Row 33 of `y` matches multiple rows in `x`.
## ℹ If a many-to-many relationship is expected, set `relationship =
##   "many-to-many"` to silence this warning.
```



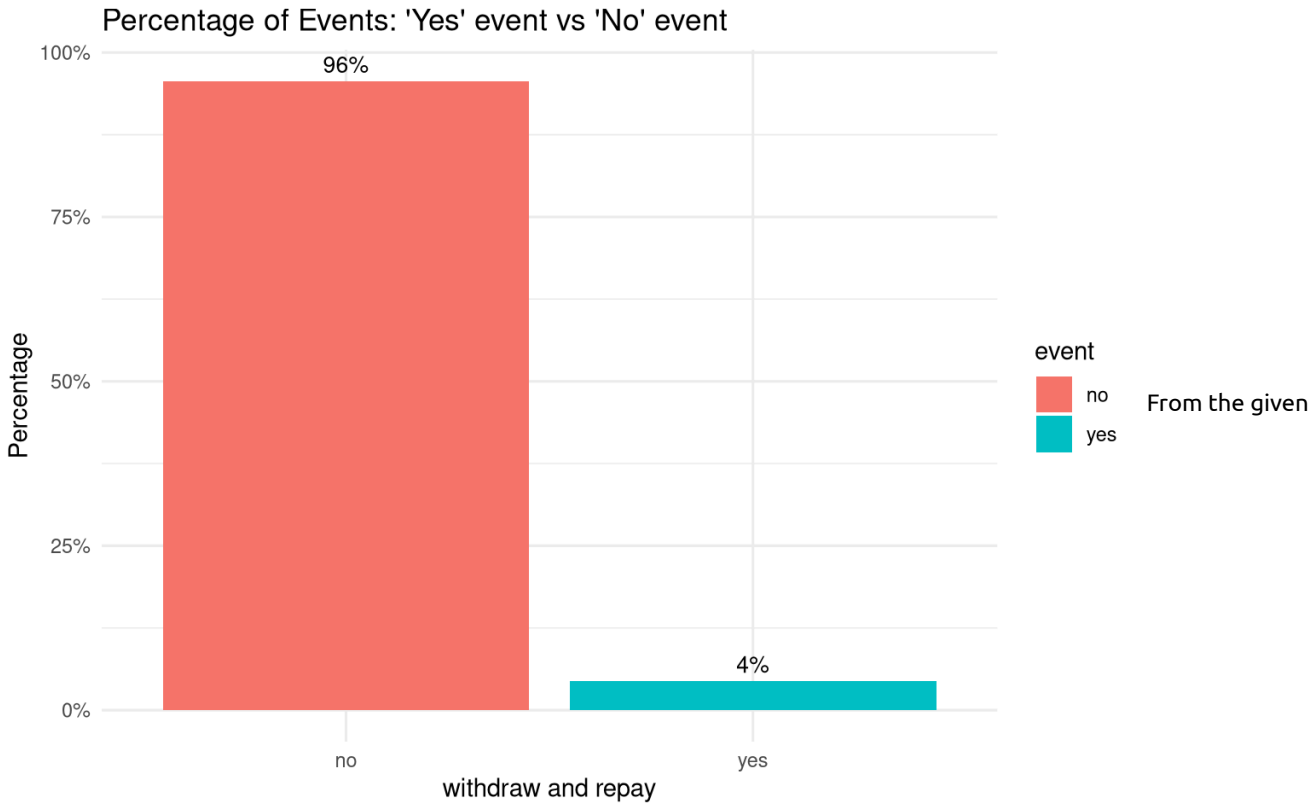

```
## Warning in inner_join(y, X, by = "id"): Detected an unexpected many-to-many relationship between `x` and
`y`.
## ℹ Row 689 of `x` matches multiple rows in `y`.
## ℹ Row 788 of `y` matches multiple rows in `x`.
## ℹ If a many-to-many relationship is expected, set `relationship =
##   "many-to-many"` to silence this warning.
```

```
## Warning in inner_join(y, X, by = "id"): Detected an unexpected many-to-many relationship between `x` and
`y`.
## ℹ Row 31 of `x` matches multiple rows in `y`.
## ℹ Row 33 of `y` matches multiple rows in `x`.
## ℹ If a many-to-many relationship is expected, set `relationship =
##   "many-to-many"` to silence this warning.
```




```
## Warning in inner_join(y, X, by = "id"): Detected an unexpected many-to-many relationship between `x` and
`y`.
## ℹ Row 26 of `x` matches multiple rows in `y`.
## ℹ Row 29 of `y` matches multiple rows in `x`.
## ℹ If a many-to-many relationship is expected, set `relationship =
##   "many-to-many"` to silence this warning.
```

```
## Warning in inner_join(y, X, by = "id"): Detected an unexpected many-to-many relationship between `x` and
`y`.
## ℹ Row 30 of `x` matches multiple rows in `y`.
## ℹ Row 33 of `y` matches multiple rows in `x`.
## ℹ If a many-to-many relationship is expected, set `relationship =
##   "many-to-many"` to silence this warning.
```



results we can find that the event percentage of those event pairs are all greater than 90%. borrow, account liquidated deposit, account liquidated deposit, borrow deposit, repay repay, account liquidated withdraw, account liquidated withdraw, borrow withdraw, repay

8 8.0 Conclusions, Limitations

From our current work, we can see that Ada boost and GBM are the best model for our binary classification. Limited to the given datasets, some of our datasets are unbalanced so that we use SMOTE to balance the trainset. It is suprising that Naive Bayes outperformed other models after SMOTE.

9 9.0 Future Work

We can try do some fine tune to our model, but due to different datasets, we probably need to do a fine tune to each event pair seperately which requires a lot of time and computation.

For classification, we can drop some uneccessary features such as “sinofYear”, “cosofYear”. Those features are highly correlated. If we can drop those co-related features, our prediction should be more precise.

10 Bibliography

Citation 1: Xiupeng Shi, Yiik Diew Wong, Michael Zhi-Feng Li, Chandrasekar Palanisamy, Chen Chai, A feature learning approach based on XGBoost for driving assessment and risk prediction, Accident Analysis & Prevention, Volume 129, 2019, Pages 170-179, ISSN 0001-4575, https://doi.org/10.1016/j.aap.2019.05.005 (https://doi.org/10.1016/j.aap.2019.05.005).