KOLEJ UNIVERSITI TUNKU ABDUL RAHMAN

FACULTY OF COMPUTING AND INFORMATION TECHNOLOGY

Assignment

BMCS2114 MACHINE LEARNING

2020/2021

Student's name/ ID Number : Nigel Lee Jian Hsee / 20WMR08882

Student's name/ ID Number : Tan Wei Siong / 20WMR08888

Student's name/ ID Number : Tan Yi Hong / 20WMR08890

Student's name/ ID Number : Tan Teoh Xin Ee / 20WMR08887

Programme : RDS

Tutorial Group : 2

Date of Submission to Tutor : 18-4-2021

Table of Contents

1.0 Introduction

As nowadays, customers are the most important assets for an organization, it is because customer retention rate is a basic requirement for a business. Without customers, there will be no business for an organization. Therefore, customer retention rates have to be alert all the time. So, this project is mainly about how a machine is able to predict customer churn for the bank. The prediction for the model will be whether the bank customer would churn or not churn. We will be using different algorithms to do the prediction of customer churn rate, so that we are able to get a better and more accurate model to do the prediction. The dataset is retrieved from the Kaggle website with the URL as (https://www.kaggle.com/sakshigoyal7/credit-card-customers). Kaggle is a dataset website which allows users to find and publish the datasets(Alex, 2019).

First, we will be looking at the problem why bank organizations need to predict the churn rate and what is the issue or reason that causes the customer churn. Second, we will get the data for our model building. Third, explore and understand the data we get such as whether the data is biased or not and the data is enough for the model to do training and so on. Fourth, carry out data processing so that we are able to help the model to train and more understand the data, which will help the model to get a better result on testing. Next, A variety of models will be used to predict the customer churn for the bank and shortlisted the models into top five best models which provide great output. All shortlisted models will fine-tune the hyperparameters using cross-validation to provide better accuracy results and try to combine them into a great solution. This is because combining the best model usually gives a better result compared to running them individually. When the final model is selected, the model performance is evaluated based on the result of the predict on the test set. Performance of each model will be evaluated based on the computational cost, accuracy, recall, precision and etc. The lower the false positive rate the better, it is because not to miss out the customer that is going to churn.

2.0 Problem Statement

Customer churn will result in a loss of revenue as well as other detrimental impacts to the bank's operation. It is vital for the bank companies to take into consideration when attempting to create long-term relationships with their customers and optimise the value of their customer base. The bank management should have their churn management which refers to the process of finding customers who are intending to switch their custom to a rival competitor. It is very hard for the bank management to know which of their customers is leaving their credit card services. With a prediction model to predict the churned customers' behaviors, the bank management will be able to predict each specific customer based on their profile. This is a solution for the bank companies to prevent their customers from churning as much as possible as they can take immediate action such as providing more benefits and better services to the predicted customers that will probably move their custom so that to retain them.

3.0 Methodology

3.1. Frame the problem and look at the big picture

Nowadays, the bank churn issue is a big topic in society. Therefore, we take up this opportunity to do this in our Machine Learning project to solve the problem. The number of churn customers is becoming more and more in the bank, by using the attribute from the dataset and the intelligence of machine learning, we are able to know the reason why the number is keep increasing and what factor could the bank officer take care or focus. Although this problem might look small, this could directly lead and affect the country's economy. Therefore, machine learning can unleash the power and let society know how strong it is.

3.2 Get the data

The dataset of bank churn is obtained from the Kaggle website. Kaggle is a big and large dataset website which provides necessary information and useful notebooks. We have chosen the recent year bank churn dataset, it is because the old bank churn dataset is not suitable to use. This is because the old dataset has not updated or maybe not affected by the recent global crisis. The dataset is just within these years so it fulfills our requirement.

3.3 Explore the data to gain insights

The dataset is given a lot of different attributes, but we do not know which one is useful and not helpful. Therefore, we have to know the importance in the relationships between pairs or small numbers of attributes. After that, we get the result of single aggregations. Then, we do a simple statistical analysis to see which one is useful and not helpful. From the analysis, we will drop the not helpful attributes. It is because it will affect the effectiveness of machine learning to train and test. From the dataset, we are also able to know that the data quality, whether is bias, the value is null value/missing value or unknown characters. Therefore, we have come out to the correlation table, and plot the graph to see the relationship in between two attributes. In the correlation table, we can see that the attributes of Customers_age and Month_on_book are highly correlated. Besides that, Credit_limit and average _open_to_buy have the highest correlated score - 0.996. Lastly, Total_trans_ct and total_trans_amt also highly correlated. The result can be based on appendix 1. Attributes with more than 0.75 skewness value (Appendix 2) were log transformed before mix max scaler to achieve better results. (Appendix 3) shows the hist plot on numerical value.

3.4 Prepare the data to better expose the underlying data patterns to Machine Learning Algorithms

After we know the useful attributes, then we will select them to use for training and testing. This is due to the reason that those useful attributes might be too much, so we have to reduce again the amount of the useful attributes. After the attributes have been selected, then we will have to look at the data. Some rows might be null value/missing value, so we have to solve the problem. Like maybe replace them to 0. Then for the categorical value we will apply one hot encoder so that it will ease the model training and testing. After cleaning the data, we will see whether data needs to merge or aggregate. This dataset is no need to construct data because the data is enough for training and testing.

3.5 Explore many different models and short-list the best ones

After the data preparation, the next stage would be to choose the actual modelling technique to implement in this project. The task needs to perform separately for each technique if there are multiple techniques are applied. Before building the model, we need to test and evaluate the model's quality and validity by generating a procedure or mechanism. In this project, we have split the data into 70% train sets and 30% test sets and build the model using the training set data. The model quality will be estimated using the test set data. The parameter of the models will be using standard parameters.

After training all the quick and dirty models from different categories, we will short-list the top 3 best models by comparing their performance. The performance is compared based on 5-fold cross validation mean accuracy score and standard deviation of the performance measure on the 5 folds. Furthermore, the training time and recall score are also essential components for estimating model performance. The recall score is very important because the higher the recall score for churn customers means the more churned customers were predicted which is also known as the true positive the higher the better. The false positive the better. This will bring a loss for the bank because the bank will miss the chance to remain the customers since it is predicted as not churn customers. Based on these criteria, top 3 models will be selected and ready for the next stage.

3.6 Fine-tune your models and combine them into a great solution

In this stage, the short-listed top 3 models will be fine-tuned to find the best parameters for the next modelling run. The model building and assessment will be iterated until we are confident with our model. Next, we will try ensemble methods. This is because combining the best models will usually perform better than running them individually. The best and confident trained model will then be saved.

3.7 Present your solution

After fine-tune the model, we will present the solution for the bank company. The presentation will let the customer know why we choose the model and the performance of the selected model.

3.8 Launch, monitor, and maintain your system

In the last phases, the model is ready for deployment. A thorough monitor and maintenance plan are required to prepare to avoid problems during the operation phase of the model. The final report of the found results are produced.

4.0 Result and Model Evaluation

4.1 Result of Dirty Train

Eight models were used for dirty training which mean train model without any parameter tuning and feature selection. Each model will be evaluated based on accuracy score, recall score, precision score, area under the curve (AUC), training time, cross validation mean score and cross validation standard deviation score. Due to the imbalance and bias dataset, the recall rate for the churn customer will be more important than other evaluation methods. Algorithm that used were Logistic Regression (LR), Gaussian Naive Bayes (GNB), Support Vector Classifier (SVC), K-nearest Neighbors (KNN), Decision Tree Classifier (DTC), Random Forest (RFC), Stochastic Gradient Descent (SGD) and XG Boost (XGB).

Model	Accuracy Score	Precision Score	Recall Score	F1 Score	AUC	Training Time (second)	CV Mean Score	CV Standard Deviation
LR	90.128%	91.394%	97.413%	93.307%	0.94	0.096	90.343%	0.514
GNB	89.207%	93.674%	93.454%	93.564%	0.89	0.012	88.585%	0.487
SVC	89.404%	89.451%	99.059%	94.010%	0.94	4.6	89.316%	0.360
KNN	86.114%	87.233%	97.766%	92.200%	0.75	0.044	85.761%	0.228
DTC	93.814%	96.026%	96.629%	96.327%	0.88	0.066	93.463%	0.496
RFC	96.347%	97.032%	98.667%	97.843%	0.99	1.0	96.238%	0.4
SGD	90.392%	92.542%	96.315%	94.391%	0.94	0.032	89.365%	1.372
XGB	97.170%	97.901%	98.746%	98.322%	0.99	0.69	97.156%	0.194

Table: 4.1.1

KNN has the worst result due to the imbalance of datasets. KNN tends to perform better on balanced datasets. The churn datasets we used almost 80% of the entries are non-churn customers. This is because KNN classify based on the distance between entries, due to the imbalance datasets many non churn customers will be classified as churn which cause the precision that low as well. The imbalance dataset also affects the performance of Logistic Regression, Support Vector Classifier and Stochastic Gradient Boosting. Most of this model classifies based on graphs. For Gaussian Naive Bayes classifier churn customer quite well with 327/488 churned customer predicted correctly because this dataset only consists of few non normal distribution and before training non normal distribution were log transform but perform quite poorly in classifying non churn customer which cause the recall lower. Tree classifiers also perform better graph based classifiers as well. XG Boost performance was the best, it is because XGBoost is a decision tree based ensemble Machine Learning algorithm that uses a gradient boosting framework. Random Forest also performs quite well because Random forest adds additional randomness to the model, while growing the trees. Instead of searching for the most important feature while splitting a node, it searches for the best feature among a random subset of features. This results in a wide diversity that generally results in a better model. Ensemble models such as Random Forest and XGBoost took longer time to train as well.

4.2 Result of Dirty Train using important variables of each model (Feature Importance on each model)

Feature Importance score is being evaluated for the eight models. Each features weighted differently for each model as they indicate the relative importance of each feature when performing a prediction. The scores are useful in a variety of situations in a predictive modelling problem, including better understanding of the data, better understanding of a model, and also most importantly it can help to reduce the number of input features. Among the 19 features, the top 8 features that have higher importance scores for each model are being selected to train the models to observe the improvement of performance.

Model	Accuracy Score	Precision Score	Recall Score	F1 Score	AUC	Training Time (second)	CV Mean Score	CV Standard Deviation
LR	89.701%	90.780%	97.648%	94.089%	0.89	0.052	89.375%	0.349
GNB	89.865%	92.497%	95.688%	94.066%	0.92	0.004	89.632%	0.466
SVC	91.477%	92.194%	98.158%	95.083%	0.94	2.5	91.587%	0.159

KNN	86.871%	88.319%	97.217%	92.555%	0.78	0.029	86.946%	0.438
DTC	93.814%	96.279%	96.354%	96.317%	0.88	0.039	94.036%	0.208
RFC	95.854%	97.014%	98.079%	97.544%	0.99	1.1	96.090%	0.185
SGD	89.404%	90.424%	97.726%	93.934%	0.89	0.021	88.901%	0.618
XGB	97.038%	97.712%	98.785%	98.246%	0.99	0.52	96.890%	0.274

Table 4.2.1

After using the selected important features to perform prediction, the improvement that can be seen was the training time has slightly decreased for almost every model. It is because the number of input features is reduced, hence the training model takes less time to train and to make a prediction. The improvement on the training time turns out to slightly lower the accuracy of all the models besides Support Vector Classifier and K Nearest Neighbour. The training time for Support Vector Classifier is cut to over half of the time and its accuracy has also increased. Same as the K Nearest Neighbour model, the training time is lessened for about 30% of the original time, as well as improvement on the accuracy by a bit. But still, the Support Vector Classifier and K Nearest Neighbour is not the best model among the others after the feature selection because the training time of Support Vector Classifier is still considered high when compared to others, whereas the accuracy of K Nearest Neighbour is only considered passable. By using feature selection, the performance of the Random Forest Classifier is decreased as it makes the accuracy dropped and with a slightly increased training time, but the overall performance of it is still considered good. The mentionable models which have good performance are the Decision Tree, Random Forest, and XGBoost. The best model among these is still the XGBoost, its accuracy only dropped by a little and it is still the highest among others, and the training time is maintained around at the middle level. While the overall performance of the other models is still almost the same before selecting the important features, it can be said that only using important variables to train the model is not improving the performance of the models by significantly, but it will shorten the training time for each model.

4.3 Result of Dirty Train (Train With One Hot Encoded on Categorical Variable)

The one hot encoder will be used to transform the categorical data into numeric arrays. This creates a binary column for each categorical variable. After encode the categorical data, eight models will be used for dirty training which mean train model without any parameter tuning and feature selection.

Model	Accuracy Score	Precision Score	Recall Score	F1 Score	AUC	Training Time (second)	CV Mean Score	CV Standard Deviation
LR	91.181%	92.004%	97.962%	94.991%	0.940	0.130	98.182%	0.560
GNB	88.253%	92.552%	93.532%	93.040%	0.890	0.011	87.568%	0.798
SVC	90.721%	91.240%	98.393%	94.681%	0.940	6.000	91.073%	0.0432
KNN	86.213%	87.351%	97.726%	92.248%	0.750	0.091	86.195%	0.219
DTC	93.485%	95.975%	96.276%	96.125%	0.880	0.078	93.819%	0.549
RFC	96.051%	96.446%	98.942%	97.678%	0.990	0.950	95.774%	0.293
SGD	91.445%	92.379%	97.883%	95.051%	0.940	0.044	90.985%	0.658
XGB	97.104%	98.012%	98.550%	98.280%	0.990	0.540	97.344%	0.186

Table 4.3.1

Based on the table above, we can see that the training time for most models has improved. This is because one hot encoder will create a binary column for each categorical variable thus there will be more columns being fit into the model. Except for Gaussian Naive Bayes, Decision Tree and Random Forest, the accuracy score for other models has slightly increased. The precision of Support Vector Classifier has dropped around 2% while other models only had small fluctuations. When coming to recall score, Support Vector Classifier, K Nearest Neighbour and XGBoost has slightly dropped. The recall score is very important for the bank company thus reducing the recall score has reduced the quality of these models. The F1 score for Gaussian Naive Bayes, Decision Tree and Random Forest has decreased. Gaussian Naive Bayes is the only model that has lower CV mean score while other has a significantly increased especially for Linear Regression. The CV Standard Deviation for Linear Regression, Gaussian Naive Bayes and Decision Tree has increased, high standard deviation means the data do not tend to be close to the mean. After thorough analysis, XGBoost is still the best model compared to others. Although there is a slight drop in recall score and take longer training time, the overall performance of XGBoost is still good and it is acceptable. The Random Forest is still performing quite well compared to others, however it’s quality has minor drop compared to training without one hot encoder.

4.4 Result of Dirty Train using important variables (Feature Importance from Correlation Table and Extra Tree Classifier)

The below table is based on the correlation table to do feature selection. From the correlation table we will select the correlated attributes and compare which attribute has the higher score. Then, add the categorical attributes.

Model	Accuracy Score	Precision Score	Recall Score	F1 Score	AUC	Training Time (second)	CV Mean Score	CV Standard Deviation
LR	89.372	90.566	97.491	93.902	0.90	0.28	89.316	0.476
GNB	89.141	91.921	95.453	93.654	0.90	0.012	88.694	0.653
SVC	88.779	88.663	99.334	93.696	0.90	6.8	88.486	0.458
KNN	86.048	87.303	97.570	92.151	0.74	0.025	86.257	0.266
DTC	88.648	93.086	93.414	93.250	0.78	0.15	89.257	0.344
RFC	93.254	93.966	98.275	96.072	0.96	3.8	92.940	0.552
SGD	89.339	91.073	96.766	93.843	0.90	0.12	89.177	0.382
XGB	92.794	93.540	98.197	95.812	0.96	1.9	92.624	0.445

Table 4.4.1: Table of correlation table feature selection

Based on the above table, we can observe that the overall model accuracy score and precision score is lower than the dirty train model accuracy score. Then, for LR, GNB, SVC and SGD models recall score is higher than the dirty train model recall score and then rest is lower than dirty train. For F1 score, LR and GNB have the higher score compared to the dirty train model and the rest is lower than the dirty train model. Auc score only LR has a higher score compared to the dirty train model and the rest is lower in score. For training time, KNN has a lower time consumed compared to a dirty train model and the rest has a higher training time. Among the model, GNB and KNN models have a higher score compared to the dirty train model and the rest are lower in score. Lastly, LR, DTC and SGD model has a lower cv standard deviation compared to the dirty train model and the rest has a higher cv standard deviation score.

As a conclusion, it is not worth doing the correlation table feature selection, because the score does not increase significantly compared to the dirty train model.

The below table is based on the extra tree classifier to do feature selection. From the extra tree classifier, we will drop the first 6 attributes with lower scores and select the rest of the attributes.

Model	Accuracy Score	Precision Score	Recall Score	F1 Score	AUC	Training Time (second)	CV Mean Score	CV Standard Deviation
LR	90.227	91.434	97.491	94.365	0.92	0.2	90.056	0.472
GNB	88.450	93.651	92.513	93.078	0.88	0.0075	87.696	0.752
SVC	89.174	89.120	99.216	93.897	0.91	4.7	88.990	0.371
KNN	86.871	88.210	97.374	92.566	0.77	0.031	86.896	0.521
DTC	93.452	96.154	96.041	96.097	0.88	0.098	93.582	0.272
RFC	96.920	96.763	98.432	97.590	0.98	1.5	95.774	0.199
SGD	89.931	91.620	96.864	94.169	0.92	0.053	88.704	1.954
XGB	96.183	96.881	98.628	97.747	0.99	0.96	96.179	0.148

Table 4.4.2: Table of extra tree classifier feature selection

Based on the table above, it indicates that most of the model does not have significant improvement after using the extra tree classifier feature selection. The accuracy score has improved for all models except for LR, KNN, and SGD. While the precision score for all models is about the same, the recall score of SVC has quite a significant improvement which increased about 3%. The SVC has improved significantly for the recall score as it is important for predicting the bank churns, and its training time has also been improved. As for the f1 score, all of the models also do not have significant changes. The mentionable of good performance models are the DTC, RFC, and XGB, these models having the higher accuracy score among the other models, while having an acceptable training time. Among these three models, XGB stands out to be the best model because of its overall performance. Although the training time has increased, the recall score and the overall score is still maintained at the higher rank among all models.

The SVC has the highest recall score but its training time is too long that makes it become the slowest model.

4.5 Results of Fine Tuning (Hyperparameter Tuning, Grid Search and Ensemble model top models)

After all the dirty trains with different conditions, the top three models were shortlisted based on the evaluation of each model and improved by applying Hyperparameter Tuning and Grid Search to find the optimal model with the optimal parameter. Short listed models were Decision Tree Classifier, Random Forest Classifier and XGBoost Model. Each model will train based differently on different data transformation such as one hot encode on categorical variables or just replace it with a label with the same set of features that picked after dirty training. The set of features were based on the importance variable of XGBoost since XGBoost meets most of the requirements such as predict churn customer well in order to prevent bank companies from miss out on their customer that is going to churn . The importance variables were quite similar for XGBoost, Random Forest and Decision Tree as well. After each Hyperparameter tuning such as one hot encode or label categorical variable, the top two models will be ensemble because combining best models might improve the performance as well. Grid Search will find out the optimal parameter combination based on the CV score. After grid search, the model will be trained and ensemble to see whether there is any improvement. The best models will be loaded and saved using joblib.

Model Result of Label Encoded and without Grid Search

Model	Accuracy Score	Precision Score	Recall Score	F1 Score	AUC	Training Time (second)	CV Mean Score	CV Standard Deviation
RFC	96.578%	97.185%	98.785%	97.978%	0.88	0.7	96.307%	0.298
DTC	93.847%	96.028%	96.668%	96.347%	0.99	0.036	93.976%	0.380
XGB	97.071%	97.824%	98.706%	98.263%	0.99	0.47	96.899%	0.469
(RFC+ XGB	97.071	98.237%	98.275%	98.256%	0.99	1.3	96.830%	0.389

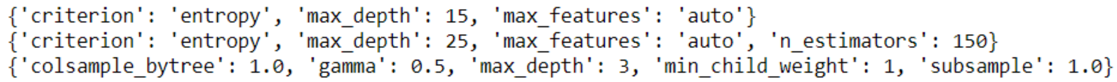
Table: 4.5.1

Model Result of One Hot Encoded and without Grid Search

Model	Accuracy Score	Precision Score	Recall Score	F1 Score	AUC	Training Time (second)	CV Mean Score	CV Standard Deviation
RFC	95.986%	96.444%	98.863%	97.638%	0.99	0.76	95.576%	0.323
DTC	93.946%	96.212%	96.590%	96.401%	0.88	0.047	93.779	0.335
XGB	97.071%	97.787%	98.746	98.264%	0.99	0.49	97.028	0.43
(RFC+ XGB	97.006	98.160%	98.275%	98.217%	0.99	1.5	96.840	0.386

Table: 4.5.2

Diagram below shows the grid search best parameter results for Decision Tree, Random Forest and XGBoost .



```
{'criterion': 'entropy', 'max_depth': 15, 'max_features': 'auto'}
{'criterion': 'entropy', 'max_depth': 25, 'max_features': 'auto', 'n_estimators': 150}
{'colsample_bytree': 1.0, 'gamma': 0.5, 'max_depth': 3, 'min_child_weight': 1, 'subsample': 1.0}
```

Diagram: 4.5.1

Result of One Hot Encoded and Train With Grid Search Parameter

Model	Accuracy Score	Precision Score	Recall Score	F1 Score	AUC	Training Time (second)	CV Mean Score	CV Standard Deviation
RFC	96.084%	96.377%	99.059%	97.700%	0.88	1.9	96.396	0.342
DTC	92.267%	95.305%	95.492%	95.398%	0.99	0.019	93.196%	0.404
XGB	96.709%	97.518%	98.589%	98.051%	0.99	0.4	97.077%	0.423
(RFC+ XGB	96.874%	97.969%	98.314%	96.141%	0.99	1.8	97.018%	0.370

Table: 4.5.3

Result of Label Encoded and Train With Grid Search Parameter

Model	Accuracy Score	Precision Score	Recall Score	F1 Score	AUC	Training Time (second)	CV Mean Score	CV Standard Deviation
RFC	96.578%	97.148%	98.824%	97.979%	0.99	2.6	95.754%	0.149
DTC	92.070%	94.942%	95.649%	95.294%	0.99	0.015	91.923%	0.336
XGB	96.644%	97.663%	98.471%	98.010%	0.99	0.55	96.978	0.332
(RFC+ XGB	96.709	97.965%	98.118%	98.042%	0.99	2.0	96.860	0.304

Table: 4.5.4

Best Model XGBoost from 4.2. Dirty Training

Actual Churn	429	59
Actual Not Churn	31	2520
	Predicted Churn	Predicted Not Churn

Best Model XGBoost from 4.3. Dirty Training

Actual Churn	437	51
Actual Not Churn	37	2514
	Predicted Churn	Predicted Not Churn

Table: 4.5.5

Ensemble Model (Label Encode without Grid Search)

Actual Churn	443	45
Actual Not Churn	44	2507
	Predicted Churn	Predicted Not Churn

Ensemble Model (One Encode without Grid Search)

Actual Churn	441	47
Actual Not Churn	44	2507
	Predicted Churn	Predicted Not Churn

Table: 4.5.6

Ensemble Model (Label Encode with Grid Search)

Actual Churn	436	52
Actual Not Churn	43	2508
	Predicted Churn	Predicted Not Churn

Ensemble Model (One Hot Encode without Grid Search)

Actual Churn	436	52
Actual Not Churn	48	2503
	Predicted Churn	Predicted Not Churn

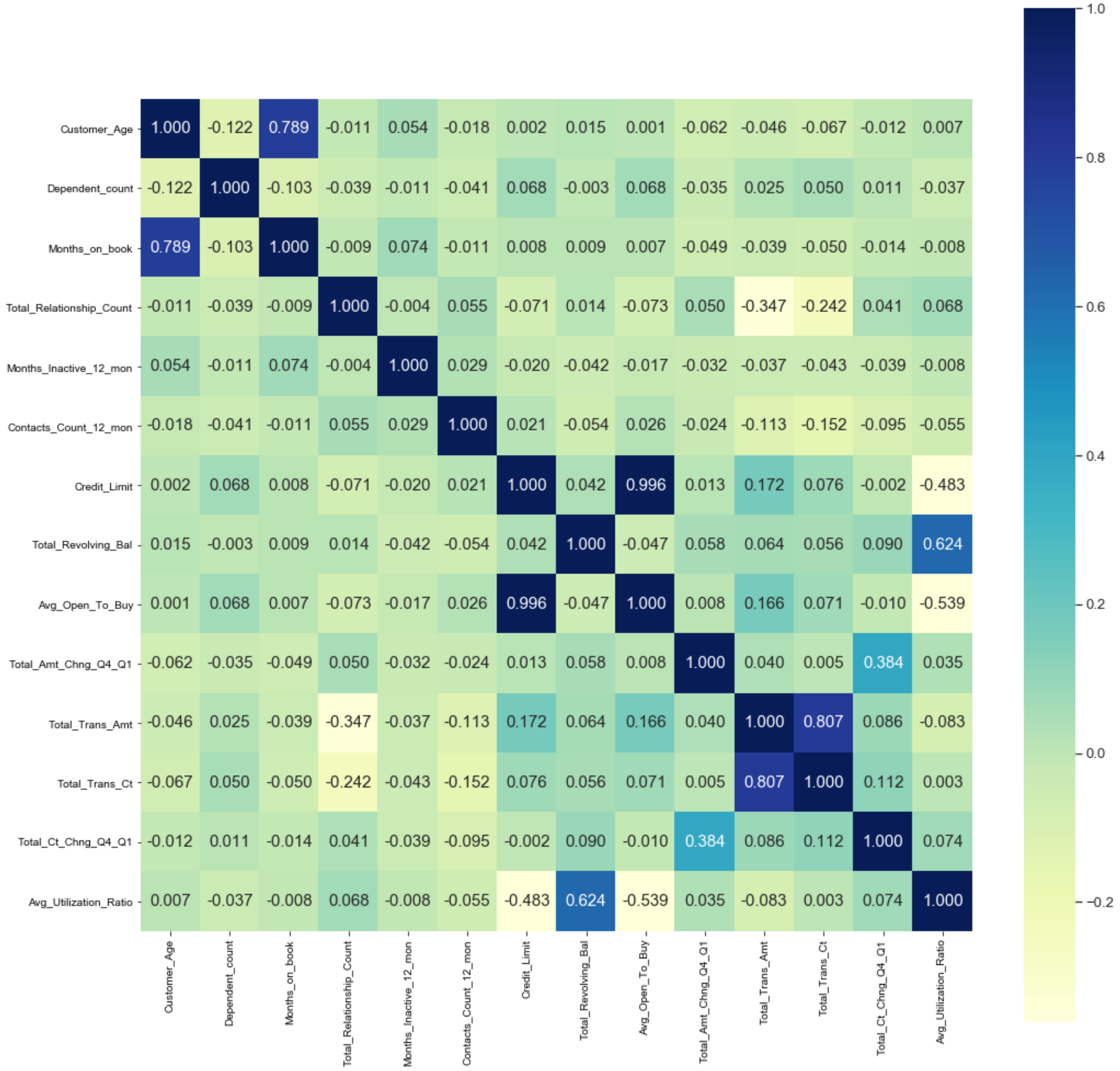
Table: 4.5.7

Based on all the ensemble models of Random Forest and XGBoost, both Ensemble Model without grid search parameter seems to predict actual churn customer better than both ensemble model with grid search parameter. On the other hand, the CV Mean Score for ensemble model with grid search parameter are slightly higher than ensemble model without grid search. This due to grid search will return the parameter based on the best cv mean score. After using grid search parameters, most of the model cross validation scores improve and the standard deviation decreases as well but slightly poor in predicting churn customer and slightly better in predicting non churn customer. In conclusion, ensemble models that label encoded and without grid search parameters predict customers which are going to churn most accurately with 436/488 churned customers compared to all models including dirty models and fine tune models and also the highest recall rate or true positive rate for churned customers. Hence this model suits and meets our requirement which detects as many churn customers as possible and does not want to miss out customers that are going to churn.

5.0 Task Allocation

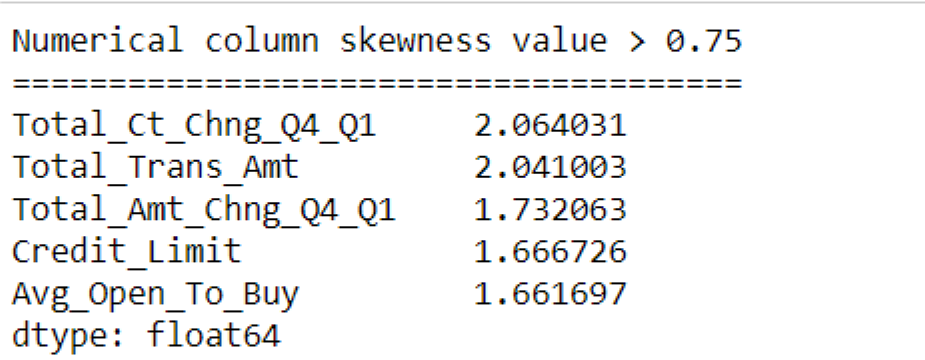
Name	**Task**
Nigel Lee Jian Hsee	-EDA -Dirty Train (Train without One Hot Encoded on Categorical Variable) -Fine Tuning (Hyperparameter Tuning + Grid Search)
Tan Wei Siong	-EDA -Dirty Train (Train With One Hot Encoded on Categorical Variable)
Tan Yi Hong	-EDA -Dirty Train (Using Important Variables of each model)
Tan Teoh Xin Ee	-EDA -Dirty Train (Train Using Extra Tree Importance Variable + Drop High Correlated Variable)

Appendix



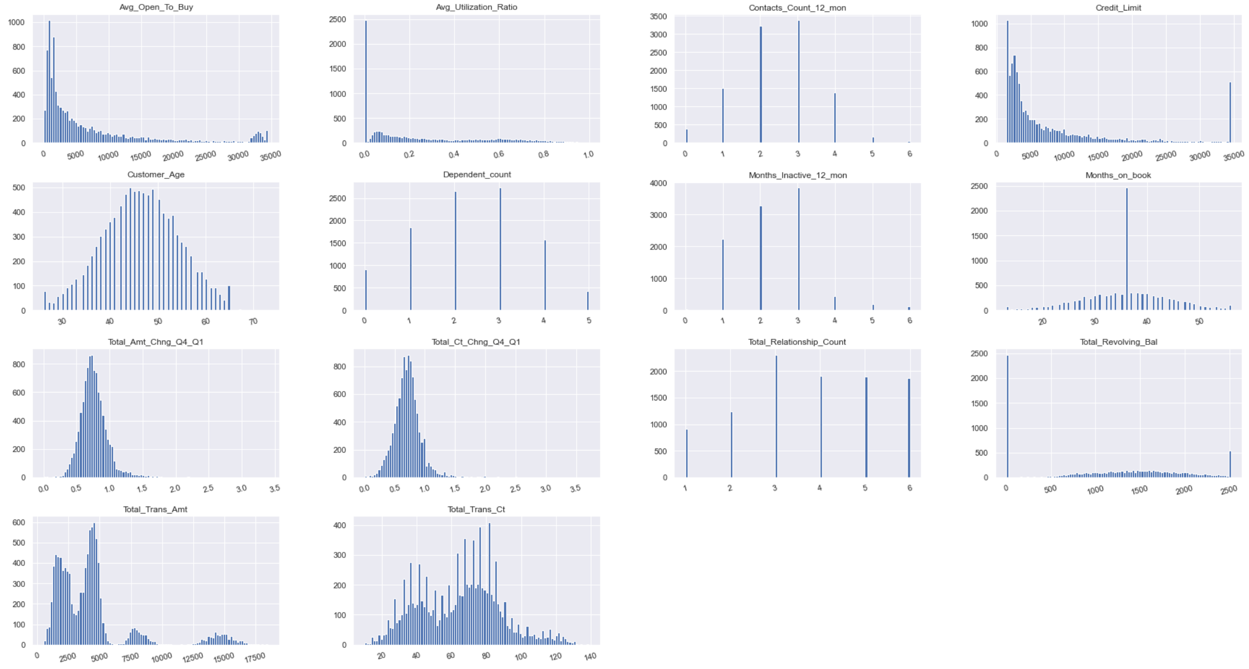
	Customer_Age	Dependent_count	Months_on_book	Total_Relationship_Count	Months_Inactive_12_mon	Contacts_Count_12_mon	Credit_Limit	Total_Revolving_Bal	Avg_Open_To_Buy	Total_Amt_Chng_Q4_Q1	Total_Trans_Amt	Total_Trans_Ct	Total_Ct_Chng_Q4_Q1	Avg_Utilization_Ratio
Customer_Age	1.000	-0.122	0.789	-0.011	0.054	-0.018	0.002	0.015	0.001	-0.062	-0.046	-0.067	-0.012	0.007
Dependent_count	-0.122	1.000	-0.103	-0.039	-0.011	-0.041	0.068	-0.003	0.068	-0.035	0.025	0.050	0.011	-0.037
Months_on_book	0.789	-0.103	1.000	-0.009	0.074	-0.011	0.008	0.009	0.007	-0.049	-0.039	-0.050	-0.014	-0.008
Total_Relationship_Count	-0.011	-0.039	-0.009	1.000	-0.004	0.055	-0.071	0.014	-0.073	0.050	-0.347	-0.242	0.041	0.068
Months_Inactive_12_mon	0.054	-0.011	0.074	-0.004	1.000	0.029	-0.020	-0.042	-0.017	-0.032	-0.037	-0.043	-0.039	-0.008
Contacts_Count_12_mon	-0.018	-0.041	-0.011	0.055	0.029	1.000	0.021	-0.054	0.026	-0.024	-0.113	-0.152	-0.095	-0.055
Credit_Limit	0.002	0.068	0.008	-0.071	-0.020	0.021	1.000	0.042	0.996	0.013	0.172	0.076	-0.002	-0.483
Total_Revolving_Bal	0.015	-0.003	0.009	0.014	-0.042	-0.054	0.042	1.000	-0.047	0.058	0.064	0.056	0.090	0.624
Avg_Open_To_Buy	0.001	0.068	0.007	-0.073	-0.017	0.026	0.996	-0.047	1.000	0.008	0.166	0.071	-0.010	-0.539
Total_Amt_Chng_Q4_Q1	-0.062	-0.035	-0.049	0.050	-0.032	-0.024	0.013	0.058	0.008	1.000	0.040	0.005	0.384	0.035
Total_Trans_Amt	-0.046	0.025	-0.039	-0.347	-0.037	-0.113	0.172	0.064	0.166	0.040	1.000	0.807	0.086	-0.083
Total_Trans_Ct	-0.067	0.050	-0.050	-0.242	-0.043	-0.152	0.076	0.056	0.071	0.005	0.807	1.000	0.112	0.003
Total_Ct_Chng_Q4_Q1	-0.012	0.011	-0.014	0.041	-0.039	-0.095	-0.002	0.090	-0.010	0.384	0.086	0.112	1.000	0.074
Avg_Utilization_Ratio	0.007	-0.037	-0.008	0.068	-0.008	-0.055	-0.483	0.624	-0.539	0.035	-0.083	0.003	0.074	1.000

Appendix 1: Correlation table from bank churn dataset



```
Numerical column skewness value > 0.75
======================================
Total_Ct_Chng_Q4_Q1     2.064031
Total_Trans_Amt         2.041003
Total_Amt_Chng_Q4_Q1    1.732063
Credit_Limit            1.666726
Avg_Open_To_Buy         1.661697
dtype: float64
```

Appendix 2: Numerical column skewness value > 0.75



Appendix 3: Hist Plot on Numerical Value

Reference

Smart Vision Europe, 2021, What is the CRISP-DM methodology?, Viewed on 15 April 2021, <https://www.sv-europe.com/crisp-dm-methodology/>

Sakshi, G. , 2020, Credit Card customers, Viewed on 18 March 2021, <https://www.kaggle.com/sakshigoyal7/credit-card-customers>

Bilal, E. , 2014, How does one choose which algorithm is best suitable for the dataset at hand?, Viewed on 19 March 2021, <https://www.researchgate.net/post/How-does-one-choose-which-algorithm-is-best-suitable-for-the-dataset-at-hand>

Onel, H. , 2018, Machine Learning Basics with the K-Nearest Neighbors Algorithm, Viewed on 19 March 2021, <https://towardsdatascience.com/machine-learning-basics-with-the-k-nearest-neighbors-algorithm-6a6e71d01761>

Xgboost developers, 2020, XGBoost Documentation, Viewed on 23 March 2021, <https://xgboost.readthedocs.io/en/latest/>