Introduction to Machine Learning Through Classification

Group homework 2 : Gautier Pellerin - Fahd Rehioui

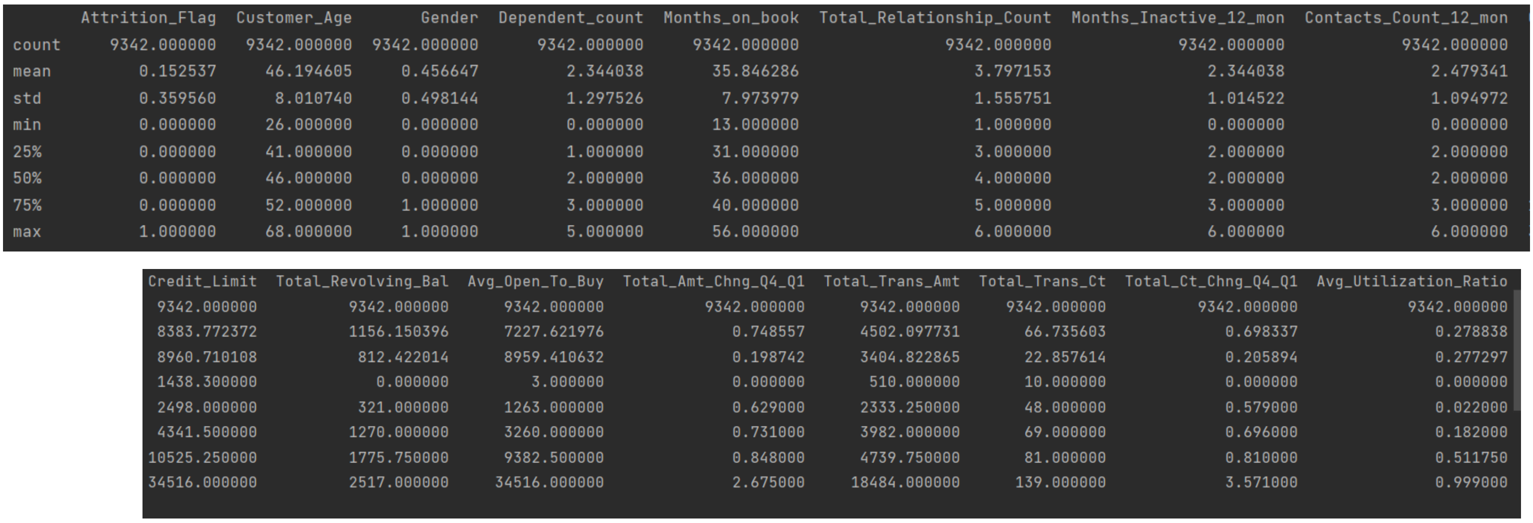
EDHEC BUSINESS SCHOOL

02 MAY 2023

Part I: Training and validation sample

Q1) Describe the data set according to the problem

First, it is important to note that the dataset contains both quantitative and qualitative variables. We used the function summary to produce the following output :



```
       Attrition_Flag  Customer_Age       Gender  Dependent_count  Months_on_book  Total_Relationship_Count  Months_Inactive_12_mon  Contacts_Count_12_mon
count     9342.000000   9342.000000  9342.000000      9342.000000     9342.000000               9342.000000             9342.000000            9342.000000
mean         0.152537     46.194605     0.456647         2.344038       35.846286                  3.797153                2.344038               2.479341
std          0.359560      8.010740     0.498144         1.297526        7.973979                  1.555751                1.014522               1.094972
min          0.000000     26.000000     0.000000         0.000000       13.000000                  1.000000                0.000000               0.000000
25%          0.000000     41.000000     0.000000         1.000000       31.000000                  3.000000                2.000000               2.000000
50%          0.000000     46.000000     0.000000         2.000000       36.000000                  4.000000                2.000000               2.000000
75%          0.000000     52.000000     1.000000         3.000000       40.000000                  5.000000                3.000000               3.000000
max          1.000000     68.000000     1.000000         5.000000       56.000000                  6.000000                6.000000               6.000000
```

```
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
 9342.000000          9342.000000      9342.000000           9342.000000      9342.000000     9342.000000          9342.000000            9342.000000
 8383.772372          1156.150396      7227.621976              0.748557      4502.097731       66.735603             0.698337               0.278838
 8960.710108           812.422014      8959.410632              0.198742      3404.822865       22.857614             0.205894               0.277297
 1438.300000             0.000000         3.000000              0.000000       510.000000       10.000000             0.000000               0.000000
 2498.000000           321.000000      1263.000000              0.629000      2333.250000       48.000000             0.579000               0.022000
 4341.500000          1270.000000      3260.000000              0.731000      3982.000000       69.000000             0.696000               0.182000
10525.250000          1775.750000      9382.500000              0.848000      4739.750000       81.000000             0.810000               0.511750
34516.000000          2517.000000     34516.000000              2.675000     18484.000000      139.000000             3.571000               0.999000
```

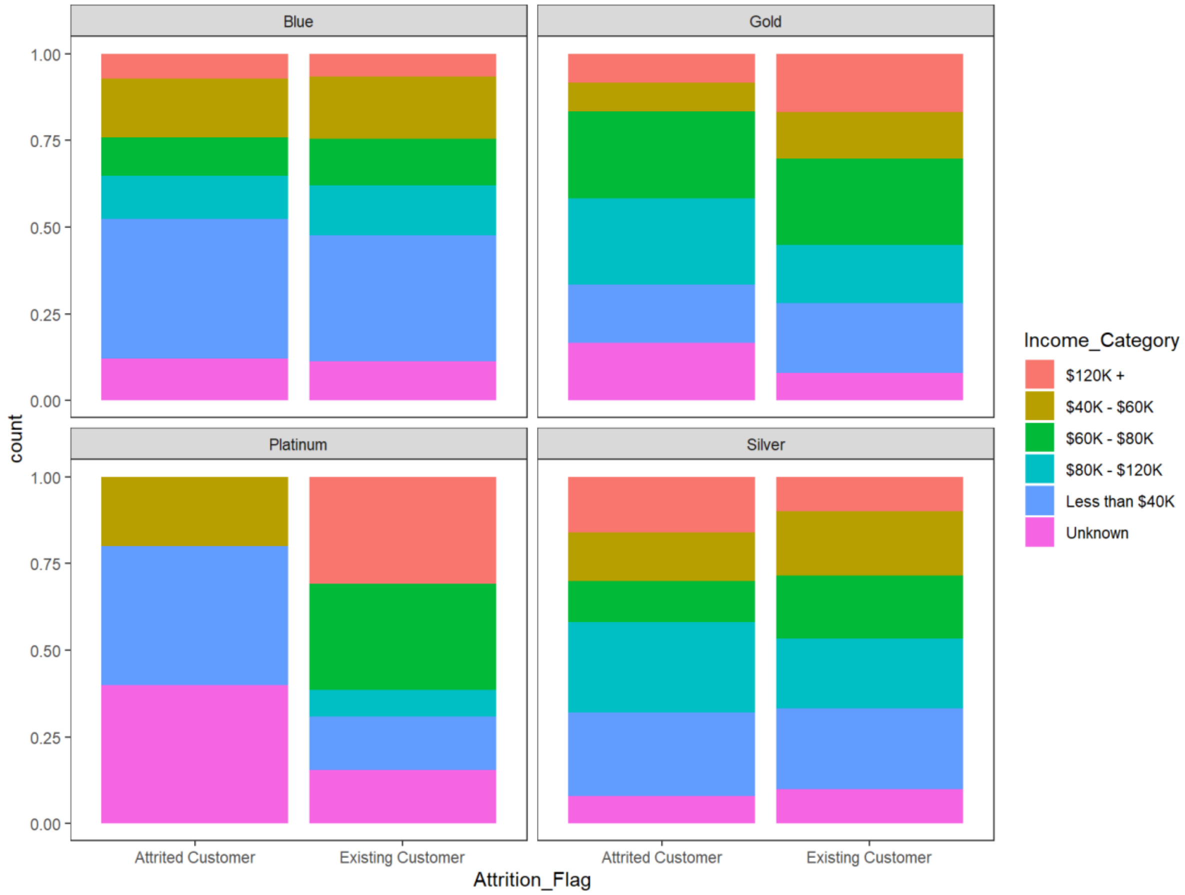
<u>Figure 1:</u> Output of summary(dataset)

The quantitative variables cannot be observed on the pictures. To perform better operation on the dataset, we decided to change all quantitative variables into dummies variables in a function called « create_variables ». In this function and in our code overall, we will manipulate mostly dataframe as it is the most useful type in python when dealing with large samples of data. We used the function get_dummies from pandas library to transform our variables. The target variable, ‘Attrion_Flag’, has also been transformed into a binary (0-1) variable as well.

With those transformation of our data, we realised we needed a variable selection procedure. We tried factorial analysis but without success, then went for an elastic net a backward selection technique.

Q2) Visualize the data set according to the problem

To get a better understanding of the dataset, we can visualize the characteristics that differentiate between the attrited and existing customers. To begin with, we can try to visualize the relationship between categorical variables. We used the proportion of the income category in attrited and existing customers within different card category which give us the following result :

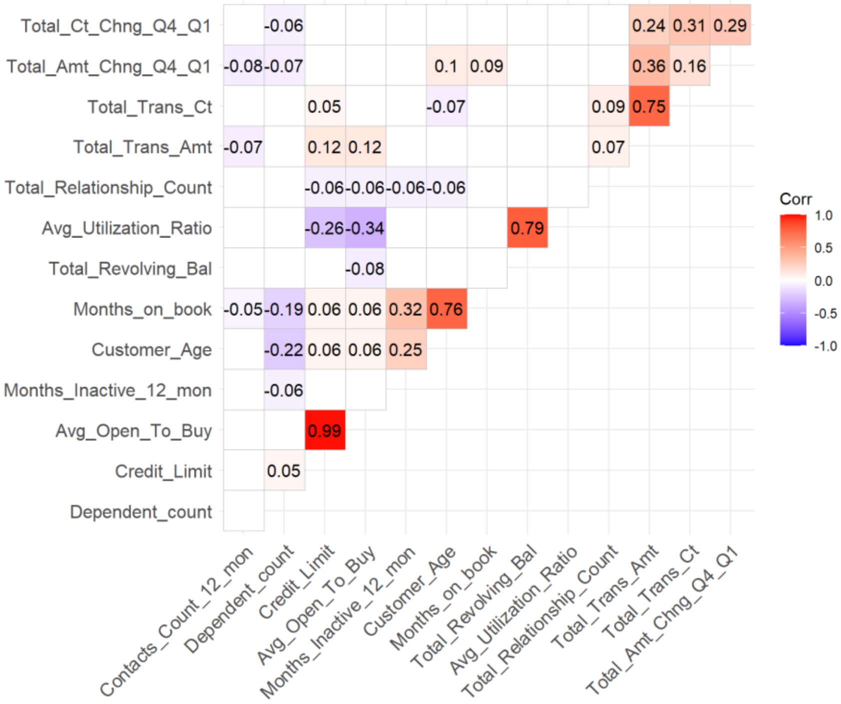


The income proportion is almost similar for existing customers and attrited customers with Blue and Silver cards. However, for Gold cardholders, attrited customers have a higher proportion of unknown income and an income range of $80K - $120K compared to existing customers, which may be a sign that more customers of these income categories are unhappy with the services related to the gold card than the one that are satisfied.

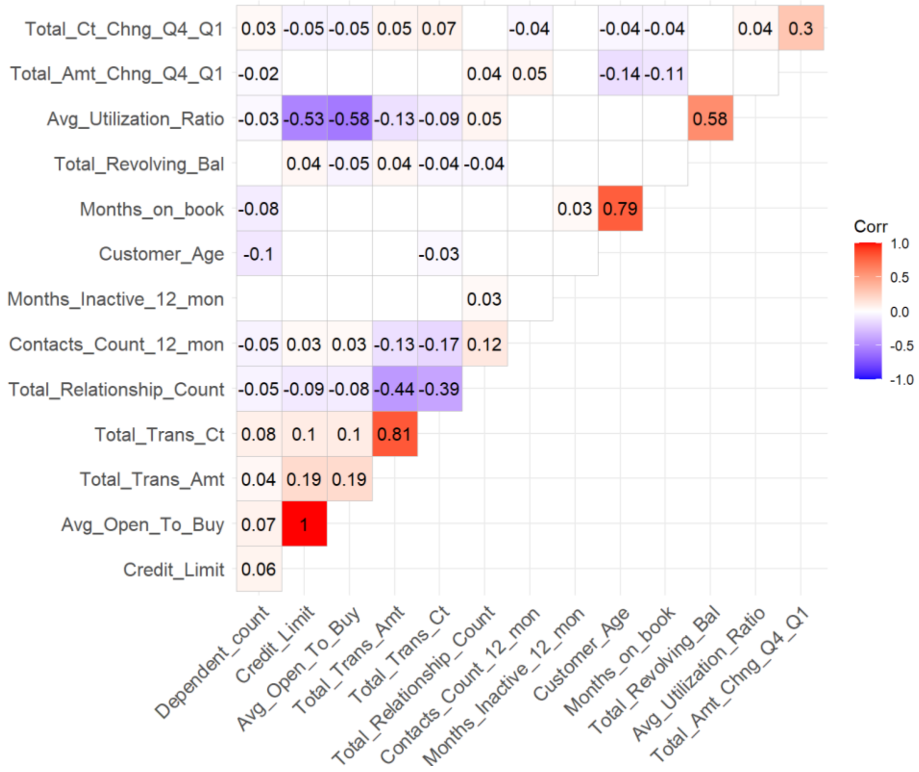
Additionally, there are no existing customers with incomes in the $40K-$60K range holding Platinum cards, while attrited customers have these missing incomes range. However, those earning more than $120K or between $60K-$80K are absent within attrited customers. This may suggest that people with incomes over $120K and between $60K-$80K are satisfied with the Platinum category, unlike those in the $40K-$60K range.

We can also create a correlation matrix heat map between numerical variables to try identify relevant variables. Here's the resulting correlation matrix heat map :

Attrited customers :



Existing customers :



In both cases we notice a potential correlation between the following variables :

- Credit limit and the average open to buy variable : These two variables are expected to have a high correlation, as they both relate to the amount of credit available to the customer

- Total transaction ct and and the total transaction amount : It is expected that there would be a positive correlation between these two variables, as the more transactions a customer makes, the higher the total transaction amount would be.

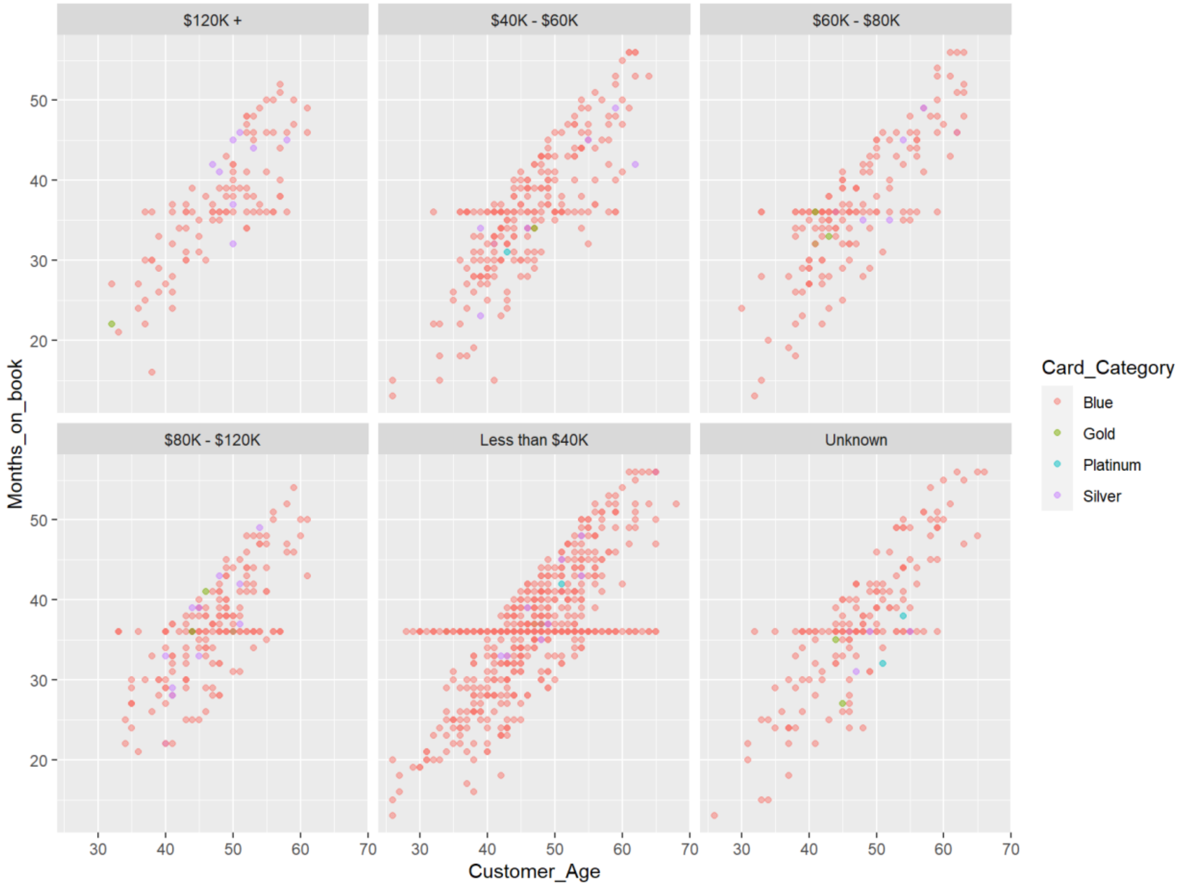
- Month on books and customer age : This means customers tend to stay with the bank for a longer period as they age.

- Avg utilization ratio and total revolving balance : This means customers who have higher debt levels also tend to have a higher utilization of their credit.

It is important to note that the correlation factors can explain the relationship between several variables, and therefore help us understanding which variables have significant impact on the others. However, the correlation matrix doesn’t tell us which variable directly influence our target variable. Therefore this whole process is more about understand the dynamics of the dataset, and maybe the crucial variables, but not a variable selection technique.

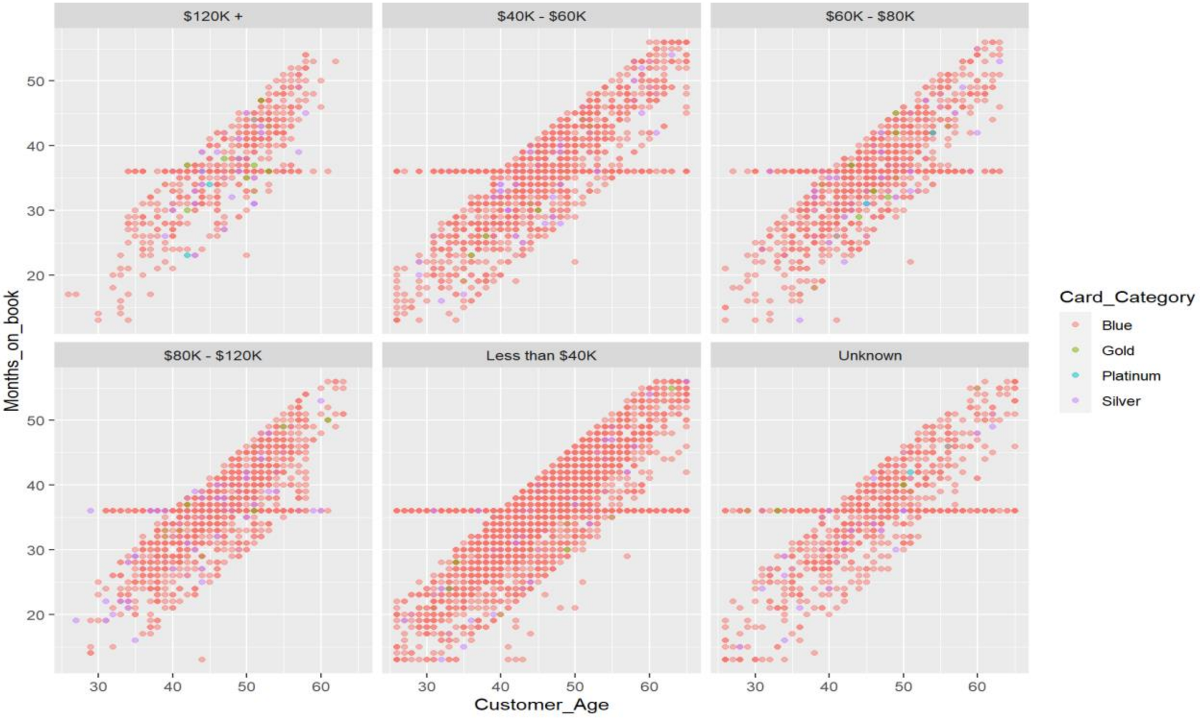
The unexpected correlation coefficient between the variables "customer age" and "months on books" caught our attention and we decided to take a closer look with the following graphs :

Attrited customers :



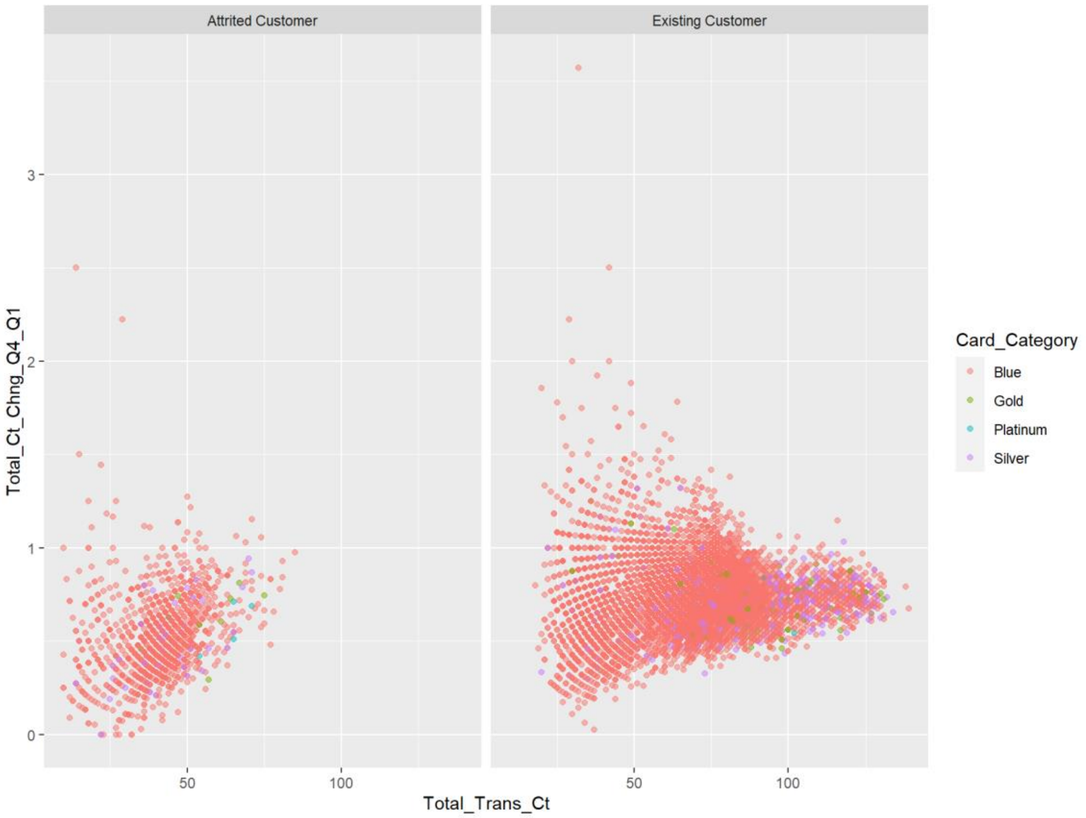
It was observed that older customers tend to exhibit higher "months on book," indicating greater loyalty as expected. Additionally, it was found that the majority of attrited customers had an annual income of less than 40K, while all age groups were represented across different income levels.

Existing customers :



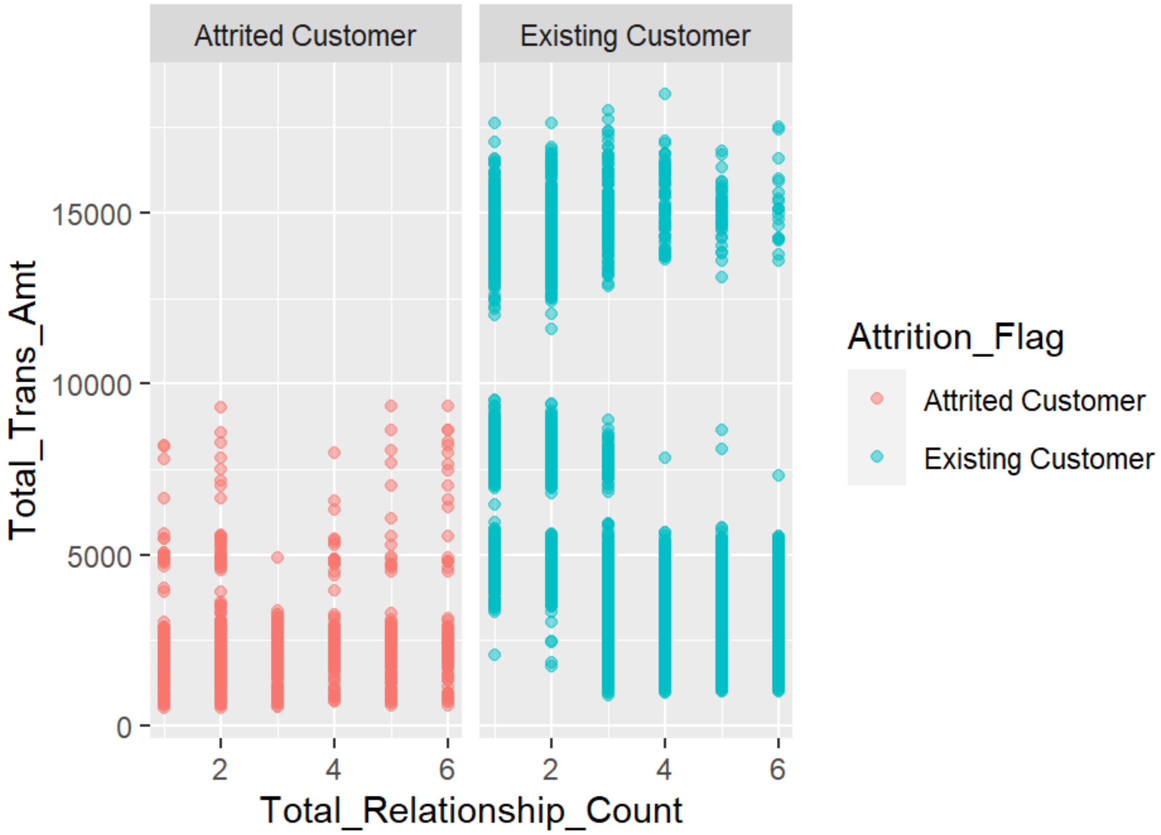
Similar to the findings for attrited customers, existing customers also exhibited the same trend as expected. This suggests that the business manager may need to focus on retaining the older clients, who are likely to be more loyal. However, it does not provide any insight into whether it can be considered a predictive variable for churning customers.

Therefore, exploring the relationship between other variables could be insightful, even if there appears to be no clear correlation as depicted by the correlation matrix heat map. For instance, investigating the relationship between the "Total_Trans_Ct" variable, which represents the total number of transactions made by the customer in the past 12 months, and the "Total_Ct_Chng_Q4_Q1" variable, which represents the percentage change in the total number of transactions between quarters, may yield valuable insights :



It was observed that the attrited customers had a lower total number of transactions compared to existing customers and mostly combined with a small “Total_Ct_Chng_Q4_Q1” variable. Additionally, the smallest total number of transactions was found in the Blue and Silver categories , which could suggest that lower transaction amounts may be indicative of potential attrition in these categories.

It may be worthwhile to explore whether there is a relationship between the "Total_Trans_Amt" variable, which represents the total dollar amount of transactions made by the customer in the past 12 months, and the "Total_Relationship_Count" variable, which represents the total number of fiancial products held by the customer :



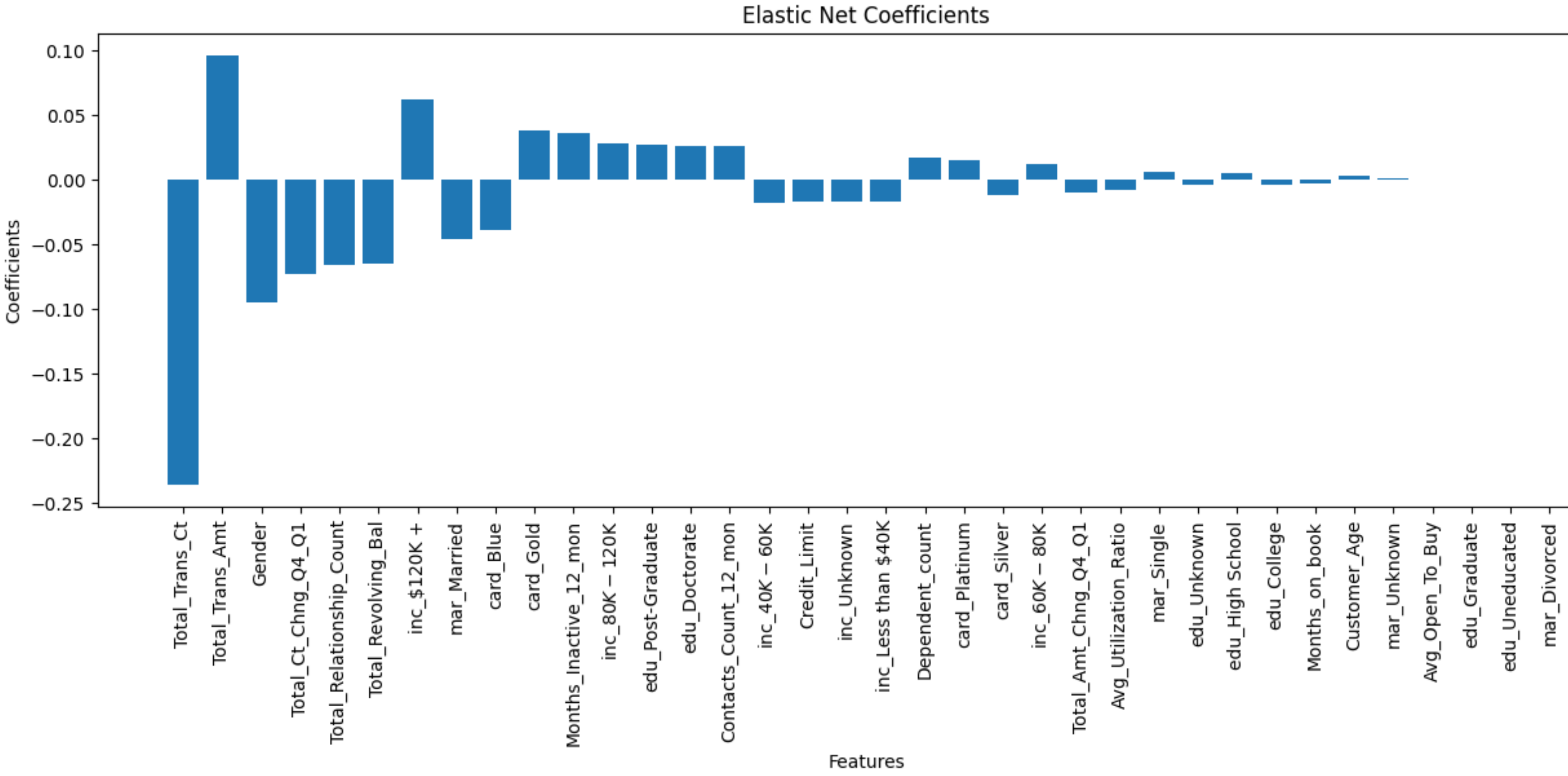
This graph reveals that attrited customers generally had total transaction amounts under $10,000, irrespective of the number of financial services they have with the credit card company. Conversely, existing customers had higher transaction amounts, primarily within the relationship count category of less than 5.
These findings suggest that the total transaction amount may be a better indicator for predicting potential attrition than the relationship count.

Q3) Develop machine learning models using the training (respectively, validation) sample. Comment the main results.

Variables Selection Process

For each of the models we will use, we will only use a subset of all the variables we defined in Q1. The main goal of this strategy is to reduce overfitting : we could have run all models with all variables but it would have included noise. By selecting the "most relevant" subset of variables, we hope to prevent overfitting.

To select this subset, we first performed an Elastic Net regression.
We divided the training & validation sample into 3 sub samples : train_sub, valid_sub and test_sub.
We then applied a scaling method to each of those samples. To find the best values of alpha and l1_ratio (our penalty terms), we ran a cross-validation with 10 folds. We then run the elastic net with the best hyper-paremeters we just found on the train_subsample and collect the coefficient of each variable. We sort them by absolutes values and print them to obtain the following picture :



As we can observe, there seems to be between 5 to 7-8 "significant" variables, after which the coefficient starts to be very small (< 0.05 in abs value). Yet not statiscally insignificant, we decided that we wanted to keep a rather simple model and therefore considered only the "best variables". We concluded that we should try our different models with a subset of the 5 top 7 or 8 most significant variables to get a relevant prediction without overfitting.

To ensure the Elastic Net is working well on new data, we run the regression once again but on the test_sub. We compute the R² and compare it to the R² of the valid_sub : 0.41 vs 0.42, meaning the regression explains 42% of the total variance. The R² on the training sample was 0.43. This value may seem quite low but it doesn't matter that much for our goal to identify the best variables.

In fact, a low R² value for an elastic net model does not necessarily mean that the model's coefficients are not useful for variable selection. The coefficients represent the estimated effect of each independent variable on the dependent variable, regardless of the overall R² value of the model. Therefore, even if the model does not explain a large proportion of the variance, the coefficients may still be useful for identifying the most important variables.

Another method we used for variable selection is backward selection with BIC criterion. This runs a logistic regression and reduces the number of columns based on the bayesian information criterion. We iterate over the columns and look for the n columns with the smallest BIC criterion. At the end, our function returns the final subset of minimal BIC criterion, which will be the columns we could use for our models.

It is interesting to note that some of the "best" variables are found by both elastic net and backward selection methods, while they disagree with some others.
In fact, out of the best 6 variables, 3 are shared : 'Total_Trans_Ct' , 'Total_Ct_Chng_Q4_Q1' and 'Total_Relationship_Count' are defined by both method as one the most crucial variables to explain attrition or not.
When we look at what those variables represent, we can easily make an interpretation of those result :

- The more a customer used it's card in the last four quarter, the less likely he will want to change it's provider as it's used to it and needs it very often, we can imagine that he cannot afford to "waste" time to get a new card and rather stay with its own provider : this refers to the 'Total_Ct_Chng_Q4_Q1' and ''Total_Trans_Ct' variables.
- The more a customer has financial services (aka relations) with the credit card provider, the "least" the customer should change provider because of the inconvenience it may generate : this refers to the ''Total_Relationship_Count' ' variable.

Model testing

For each of the family model that we will present, we have done the following procedure :

1) Split the data into training and validation sample
2) If needed for the model, rescale the quantitative variables to not overweight some variables
3) Select a “subset” of the variables found via elastic net or backward selection
4) Run a cross-validation method to get the best hyperparameters based on a metrics (roc_auc & recall)
5) Train a model with the training sample
6) Do a cross-validation with k=5 or 10 folds on the training sample and note the metrics chosen
7) Evaluate the model on the validation sample. We print the precision, accuracy, recall, F1 and MSE
8) Ensure the cross validation scores are close from the results obtained in 7)
9) Start again from 3) with a larger or smaller subset of variables

10) Compare the key metrics for each of the subset and pick what seems to be the best model.

To evaluate our model performance, we will use confusion matrix based criteria : MSE, Precision, Recall, F-Measure, Accuracy and ROC AUC.
About the model comparison : the aim of the project is to effectively detect a customer that is about to leave, and offer him better services to keep him.
If the company wants to offer the best services to the customers who are predicted to churn, then **the cost of a false negative could be very high, as the company may miss out on an opportunity to retain a valuable customer**. In such cases, it may be **more important to have a high recall (even if it means sacrificing some precision) to ensure that the company is able to identify as many potential attrited customers as possible and offer them better services**.

Therefore, to select the best model among a class, we prefer the model with the highest recall. In case of almost identical results (less than 1% difference), we will pick the simplest model (Occam's Razor principle).

1st Model : Logistic Regression

It is important to remember that due to its nature, logit needs scaling on the quantitative variables. We first use the subset of variables from elastic net, which produces the following result :

Nb Var	C	Precision	Recall	F-Measure	Accuracy	MSE
4	1	0.6239	0.5015	0.5615	0.878	0.122
5	1	0.709	0.6049	0.6528	0.9016	0.0984
6	5	0.7676	0.6469	0.7021	0.916	0.0840
7	10	0.7741	0.6469	0.7048	0.9171	0.0829
8	0.1	0.7787	0.6643	0.717	0.9197	0.0803
9	0.1	0.7755	0.6643	0.7156	0.9192	0.0808

We easily see that even with up to 9 variables, we do not have a very high recall rate. Our best performing model is with 8 variables, as it has the same recall as 9 variables but is simpler by definition.

The result from the subset using backward selection :

Nb Var	C	Precision	Recall	F-Measure	Accuracy	MSE
4	1	0.7186	0.5804	0.6422	0.9010	0.0990
5	1	0.7261	0.5839	0.6473	0.9026	0.0974
6	1	0.7125	0.5979	0.6502	0.9016	0.0984
7	0.1	0.7322	0.6119	0.6667	0.9064	0.0936
8	0.1	0.7490	0.6259	0.6819	0.9106	0.0894

We plot the ROC curve for both the “best” backward selection and elastic net variable selection, as well as the random guess curve as a benchmark.

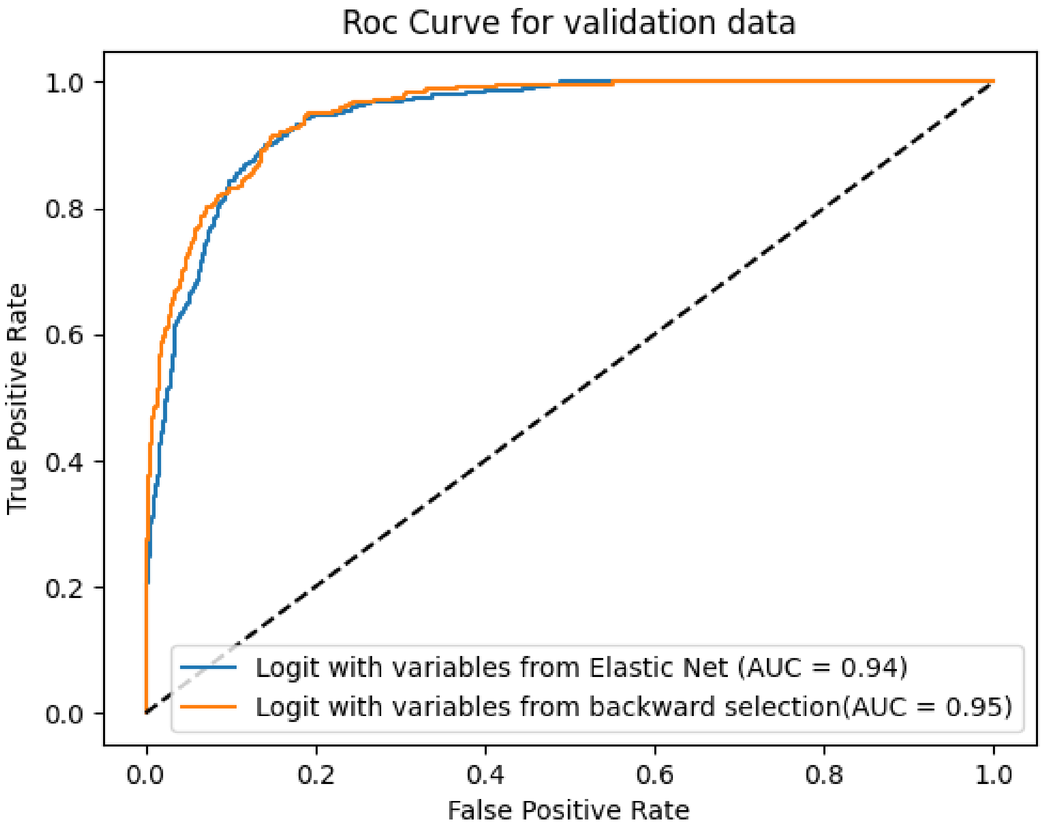


Figure 1: ROC Curve - Logit Model

We can clearly see that our models are outperforming the majority rule / random guess (both produce tpr = fpr , which is represented by the black dashed line). The AUC is tiny better for backward selection but we must remember that the recall is our main concern : it is 4% higher with the net, therefore we will use the logit model with 8 variables selectionned from the elastic net.

2nd Model : KNN

It is important to remember that due to its nature, KNN needs scaling on the quantitative variables.

For this model, we have obtained the following result on the validation sample :

Nb var	K	Precision	Recall	F-Measure	Accuracy	MSE
4	9	0.7223	0.6049	0.6784	0.9123	0.0877
5	10	0.8522	0.6853	0.7597	0.9337	0.0663
6	5	0.8525	0.7273	0.7849	0.939	0.0610
7	5	0.8394	0.7308	0.7813	0.9374	0.0626
8	5	0.8333	0.7343	0.7807	0.9369	0.0631

We can observe that the best results on the validation sample are obtained with 7 variables. However, we stated earlier that a difference in Recall below 1% would not matter if the 2nd best performing model has better metrics than the top recall performer. This is exactly the case here and we will therefore pick 6 variables.
5 variables offer almost the same result for every metrics but the Recall, so we prefer keeping our higher Recall.

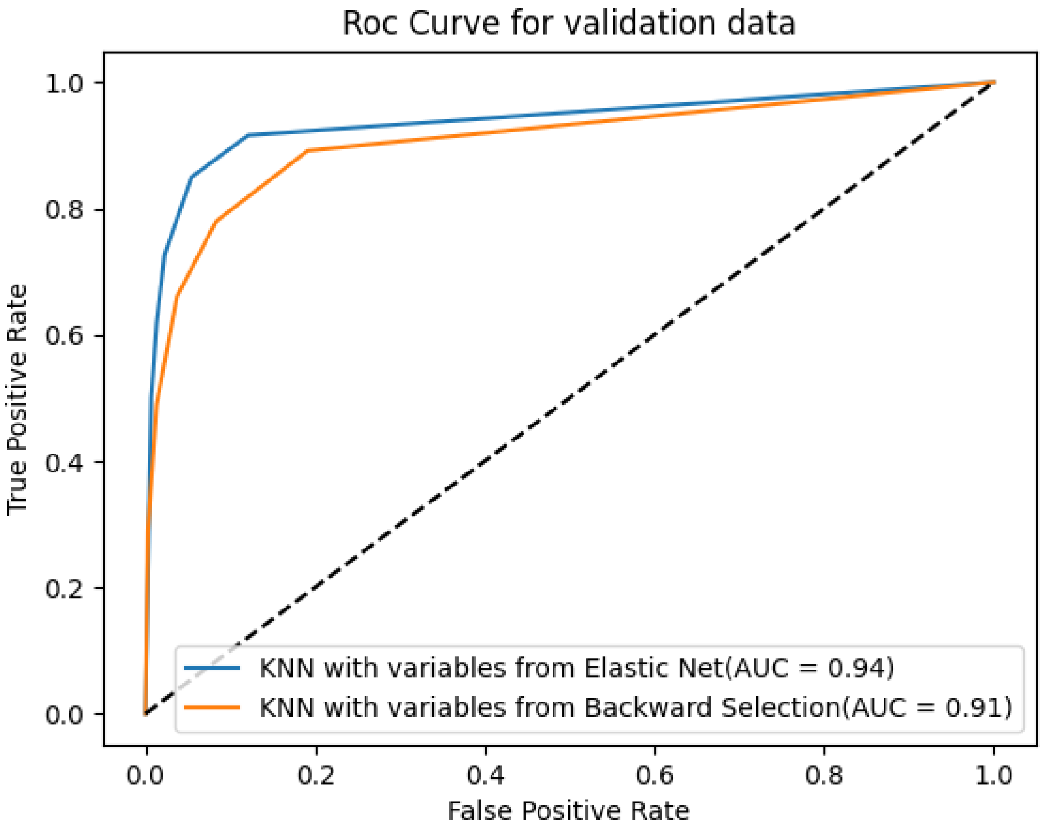


Figure 2 : ROC Curve - KNN model

We plot the ROC curve as well as the random guess curve as a benchmark. We can clearly see that our model is outperforming the majority rule / random guess.

The logit model made us think that the backward selection variable wasn’t performing as good as the elastic net selection. This result is confirmed again on knn with the above ROC Curve, where we can see the best model from backward selection performs worse than the best from elastic net selection.

3rd Model : Binary Tree - CART Model

It is important to remember that due to its nature, binary tree do not need scaling on the quantitative variables. To evaluate our model performance, we will use confusion matrix based criteria : MSE, Precision, Recall, F-Measure, Accuracy and ROC AUC.

For this model, we have obtained the following result on the validation sample :

Nb var	Depth	Split	Precision	Recall	F-Measure	Accuracy	MSE
4	6	17	0.7761	0.7273	0.7059	0.9262	0.0738
5	6	19	0.793	0.7902	0.7916	0.9363	0.0637
6	6	20	0.8571	0.7552	0.803	0.9433	0.0567
7	6	19	0.8571	0.7552	0.803	0.9433	0.0567
8	6	18	0.8571	0.7552	0.803	0.9433	0.0567

We observe the best result for 5, 6,7 or 8 variables but we must remember that the most important metrics is the Recall. The model with 5 variables has 4% more Recall than the 6 variables, so we decided to use this one. We don’t consider 7 or 8 further since 6 has the same results and is less complex.

We plot the ROC curve as well as the random guess curve as a benchmark. We can clearly see that our model is outperforming the majority rule / random guess .

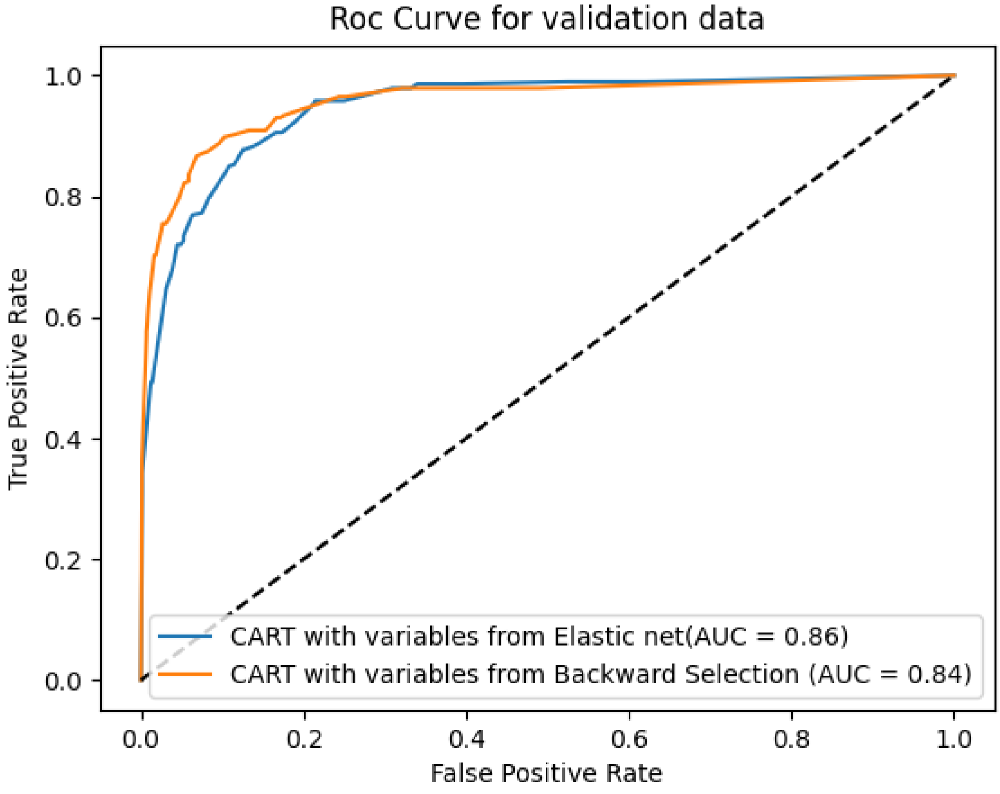


Figure 3: ROC Curve - CART model

The better performance of variable selection by elastic net is confirmed once more for the CART model, cf roc curve above.

4th Model : Random Forest

As for Binary Tree, random forest does not need scaling on the variables. To evaluate our model performance's, we will use confusion matrix based criteria : Precision, Recall, F-Measure, Accuracy and ROC AUC.
For this model, we have obtained the following result on the validation sample :

Nb Var	Split	Leaf	Estimators	Precision	Recall	F-Measure	Accuracy
4	2	4	500	0.8245	0.7603	0.7608	0.932
5	2	2	500	0.8138	0.7208	0.7542	0.9299
6	2	1	500	0.8919	0.807	0.8477	0.9556
7	5	1	500	0.8953	0.8077	0.8493	0.9561

N.B. : we do not disclose the max Depth, as hyper-parameters tuning setted to 10 for each.

We are looking for the highest recall, therefore we have to decide between 6 or 7 variables. It appears the “scores” of 7 variables model aren't significantly different from the 6 variables model. Therefore, we pick the simplest : 6 variables.

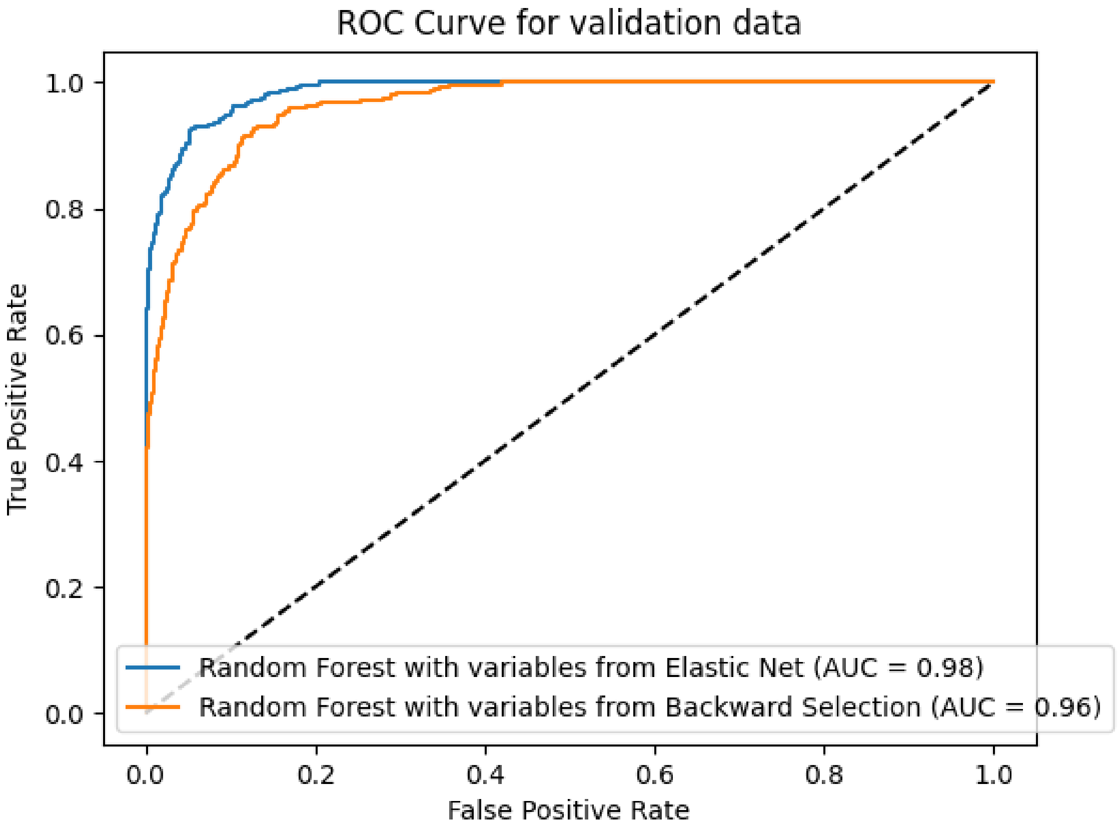


Figure 4: ROC Curve - Random Forest model

We plot the ROC curve as well as the random guess curve as a benchmark. We can clearly see that our model is again outperforming the majority rule / random guess .

Q4 : Make a comparison between the different models

As stated earlier, our main metric we want to compare is the recall rate (customers who actually leave and are correctly identified as leaving), or true positive. Then comes the precision, as we want to minimize the number of false positives : offering the best services to the customers could be by a promotion and in this case we do not want to offer promotions to customers who are not about to leave ! The F-measure is therefore an excellent choice of metric to look at when we have a trade-off between recall and precision.

Let’s review the key metrics for each of the “best” performing model within each class

Model	Nb Var	Precision	Recall	F-Measure	Accuracy	ROC AUC	MSE
Logit	8	0.7787	0.6643	0.717	0.9197	0.94	0.0803
KNN	6	0.8525	0.7273	0.7849	0.939	0.94	0.0610
CART	5	0.793	0.7902	0.7916	0.9363	0.86	0.0637
Random Forest	6	0.8919	0.807	0.8477	0.9556	0.98	0.0465

The best recall measure among all our model is the Random Forest. The logit is far from excelling at recall and has overall worse metrics than the others. The CART Model is producing almost similar results in terms of recall but is 10% below in terms of precision, our 2nd most important metric. It is also 12% below the Random Forest for the ROC AUC.

The F-measure, which is computed using both precision and recall, is also at its best for the Random Forest. Furthermore, the Random Forest also has the smallest MSE of all our models, making it the “best model” overall.

The KNN model disclose results very close from the Random Forest, with very few 1-4% below on each metrics. It would be our 2nd best model and has the advantages of being far less demanding in time complexity! We must remember the RF has 500 trees, which is a significant number and requires a longer computing time than the KNN one. In addition, the tunning of the KNN function is rather easy to do, while it requires a lot more parameters for the Random Forest. In terms of computing time, the “tuning function” we developed for the Random Forest tools between 5 to 10 min to find the optimal parameters, while less than 30 seconds were needed for the KNN tuning.

Therefore we could see the KNN as a great alternative to the RF, according to the client constraint on computing power and time limit.

Part II: Test sample

Q1 :Provide some measures of the out-of-sample performances.

We have highlighted the most important result according to us.
The Logit model produced the following out of sample result

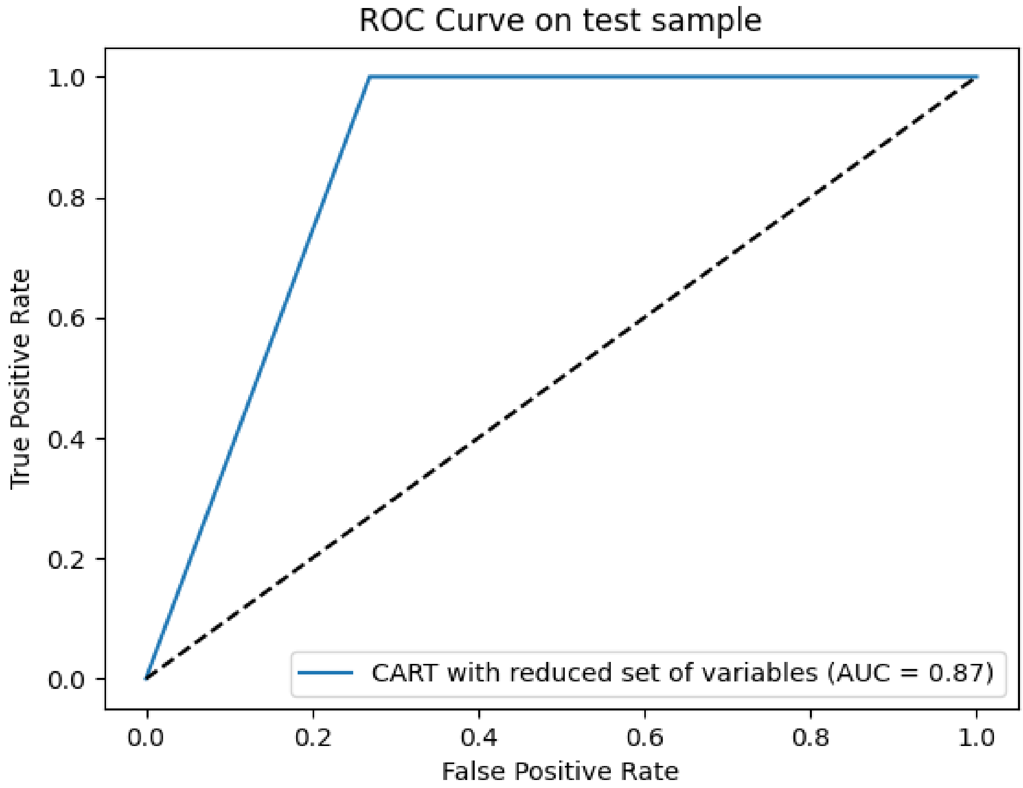


Figure 5: ROC Curve - Logit on test sample

	Precision	Recall	F1	Support
0	0.78	0.87	0.82	583
1	0.44	0.29	0.35	202
Weighted avg	0.69	0.72	0.70	785

Confusion matrix

	0	1
0	509	74
1	144	58

Accuracy	0.78	785
MSE	0.278	

KNN Model produces the following out of ample result :

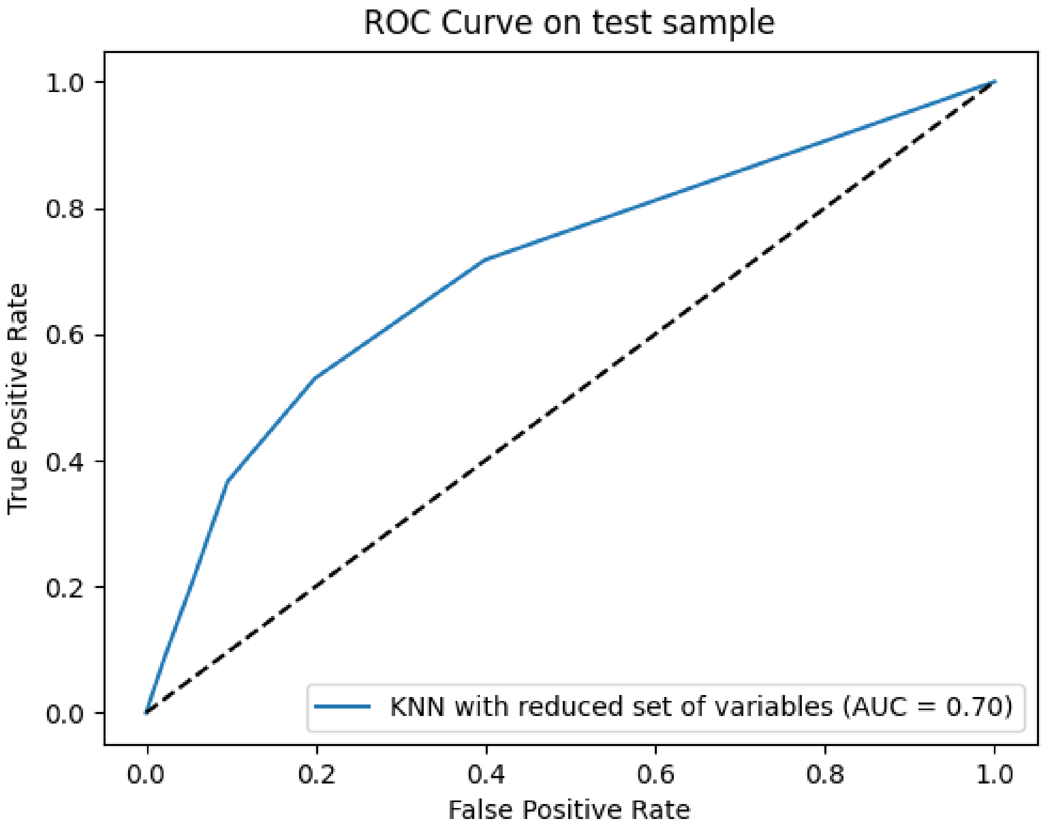


Figure 6: ROC Curve - KNN on test sample

	Precision	Recall	F1	Support
0	0.80	0.90	0.85	583
1	0.57	0.37	0.45	202
Weighted avg	0.74	0.77	0.75	785
Accuracy			0.77	785

Confusion matrix

	0	1
0	527	56
1	128	74

MSE	0.234

CART Model produces the following out of ample result :

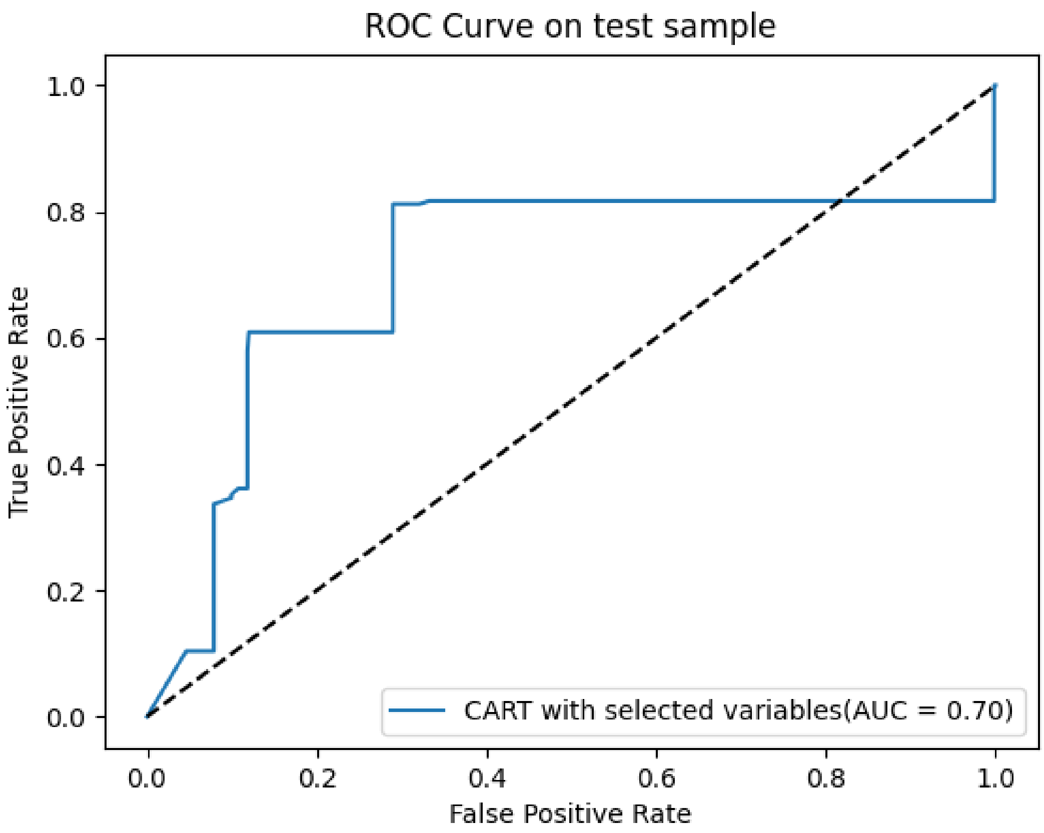


Figure 7: ROC Curve - CART on test sample

	Precision	Recall	F1	Support
0	0.76	0.92	0.83	583
1	0.43	0.17	0.25	202
Weighted avg	0.68	0.73	0.68	785
Accuracy			0.73	785
MSE			0.271	

Confusion matrix

	0	1
0	537	46
1	167	35

RF Model produces the following out of ample result :

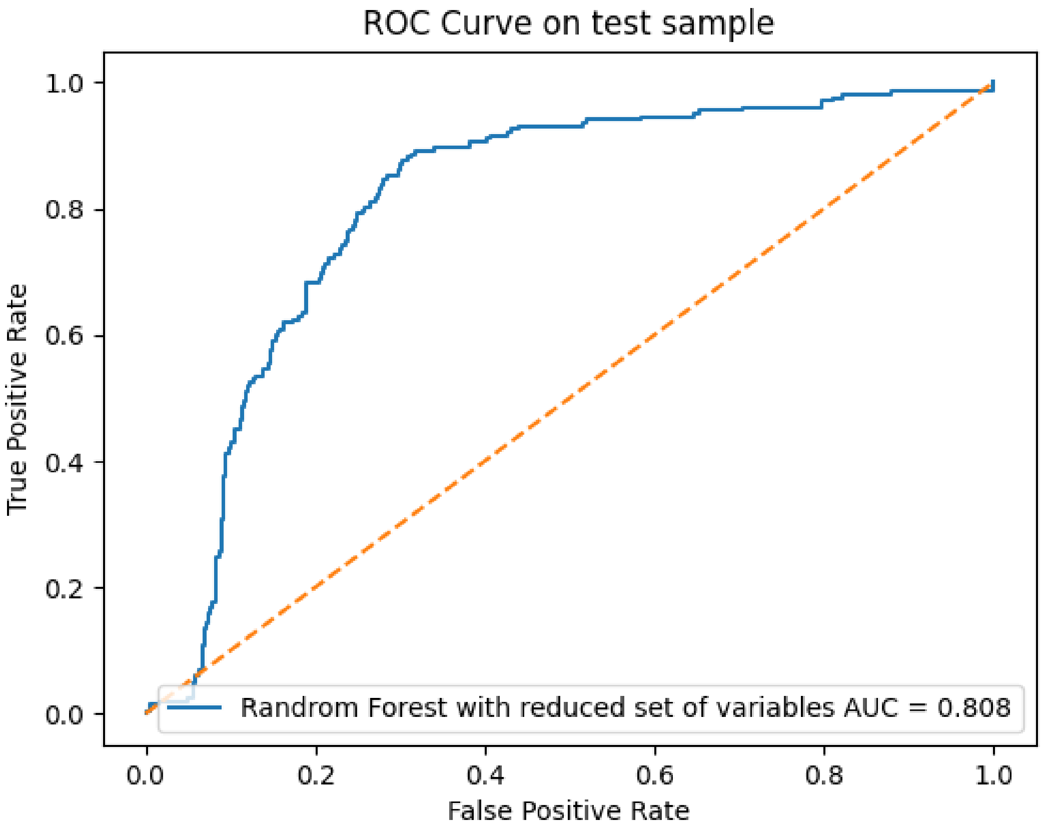


Figure 8: ROC Curve - RF on test sample

	Precision	Recall	F1	Support
0	0.83	0.88	0.86	583
1	0.59	0.49	0.53	202
Weighted avg	0.76	0.77	0.77	785
Accuracy			0.78	785
MSE			0.219	

Confusion matrix

	0	1
0	515	68
1	104	98

Q2 :Discuss and interpret your results

By looking at the ROC curve, we could think that all models performed better than a random guesser. This would implied we avoided overfitting and that the elastic net was pretty successful.

At first glance, we can observe that the weighted average recall is equal for KNN and RF models : on average, both models correctly identified 77% of the relevant instances across all classes. CART and Logit weighted recall is almost equal as well, with more than 73% of correct guesses for the two classes. The weighted Precision and therefore F1 measures are equivalent for KNN and RF, implying that both of those two models have a good balance. All models have relatively the same MSE, which is way higher than in the in-sample analysis. This obviously means our model performed much worse out-of-sample.

However we must remember that our key metric is the recall rate for class 1 ! And in this case, there are very large difference between our models. Logit and CART have very poor recall class 1 rate , below 0.3. It means those modes correctly identified fewer than 30% of the positive instances in the class 1.
KNN and RF are doing better, with respectively 37% and 49% of correctly identified positive samples for class 1. As this is what the whole project and model is about, this is the true metric we must focus on when making a recommandatio.

If the best model is only 49% right when predicting the positive cases, we can say our best model performs roughly the same as a random guesser for this class. This is obviously not a very good result and means we could improve our model.

Q3 :Which model would you recommend at the end?

We would recommend using the Random Forest. The model performed better than the other 3 models in both in-sample and out-of sample analysis. Even with only 0.49 recall for class 1, the high weighted average recall of 0.77 indicates that the model is performing well overall across all classes, taking into account class imbalances. Therefore, the model might still have some value even if the recall for class 1 is low, depending on the specific use case and the trade-offs between different metrics.

What is certain is that some improvements can be made, in terms of variable selection, maybe removing outliers before training, trying more models etc.