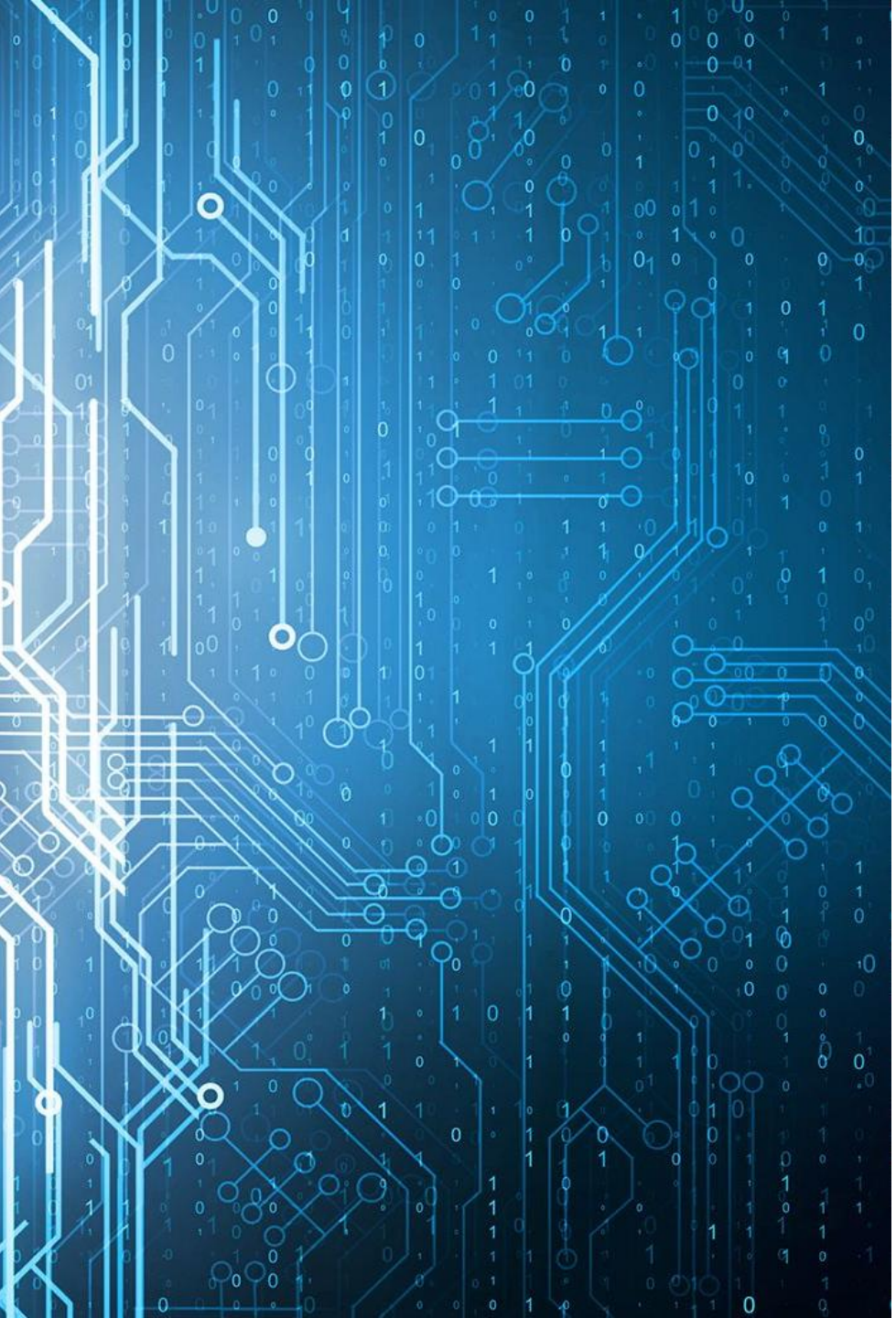


U.PORTO
FACULDADE DE CIÊNCIAS
UNIVERSIDADE DO PORTO

Introduction to Data Science

Long Assignment 2023/2024

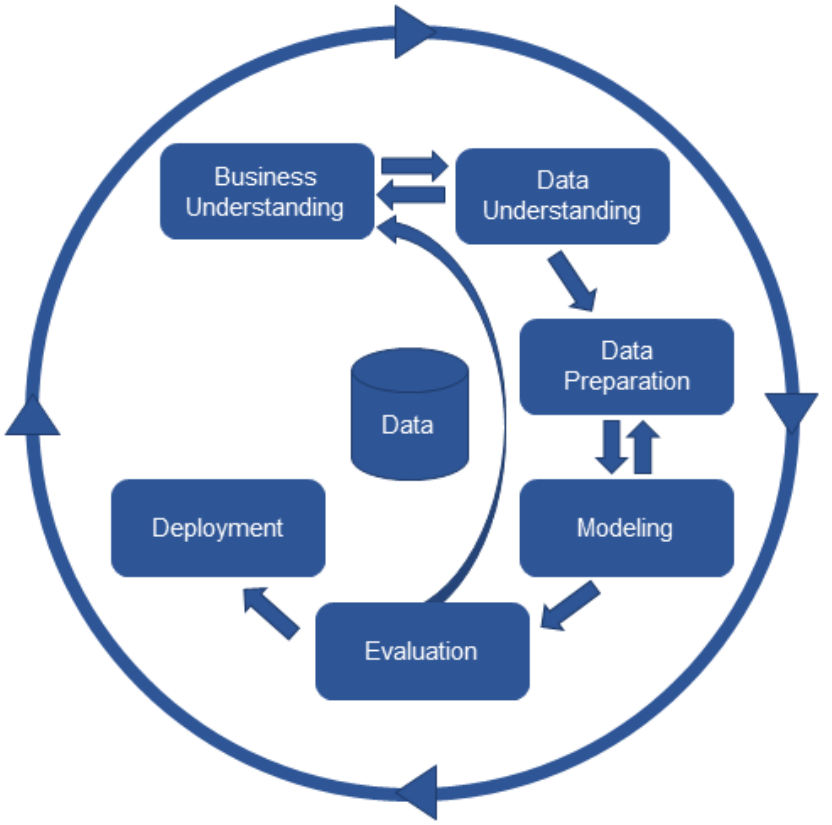
Telecom Churn

Juliana Henriques Amaral, 201907239
Maria Francisca Coelho Queirós, 201805084
Telmo Filipe Pereira Monteiro, 202308183

1. Business Understanding

Aim: develop a predictive model that can accurately identify customers at risk of churning using the CRISP-DM methodology.

- **Business objective:** telecommunications service provider want to decrease the number of costumer churn;
- **Assess situation:** dataset with information about 5000 costumers and 18 features;
- **Machine learning goals:** build a model that predicts with high precision if a costumer is going to churn or not.



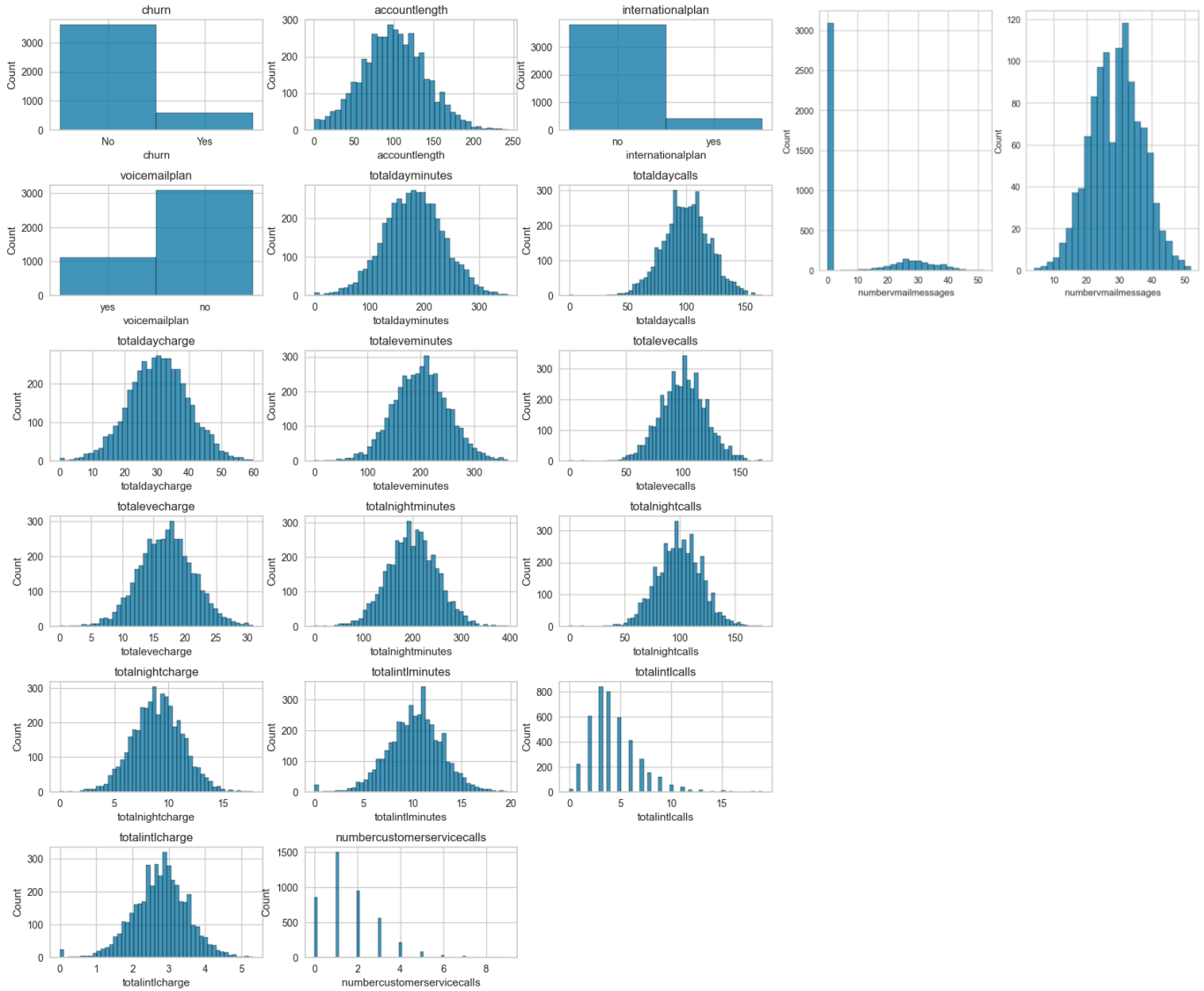
2. Data Understanding

Variable	Type	Variable	Number of missing values
churn	object	churn	0
accountlength	float64	accountlength	49
internationalplan	object	internationalplan	50
voicemailplan	object	voicemailplan	50
numbervmailmessages	float64	numbervmailmessages	50
totaldayminutes	float64	totaldayminutes	50
totaldaycalls	float64	totaldaycalls	50
totaldaycharge	float64	totaldaycharge	50
totaleveminutes	float64	totaleveminutes	50
totalevecalls	float64	totalevecalls	50
totalevecharge	float64	totalevecharge	50
totalnightminutes	float64	totalnightminutes	50
totalnightcalls	float64	totalnightcalls	50
totalnightcharge	float64	totalnightcharge	50
totalintlminutes	float64	totalintlminutes	50
totalintlcalls	float64	totalintlcalls	50
totalintlcharge	float64	totalintlcharge	50
numbercustomerservicecalls	float64	numbercustomerservicecalls	50

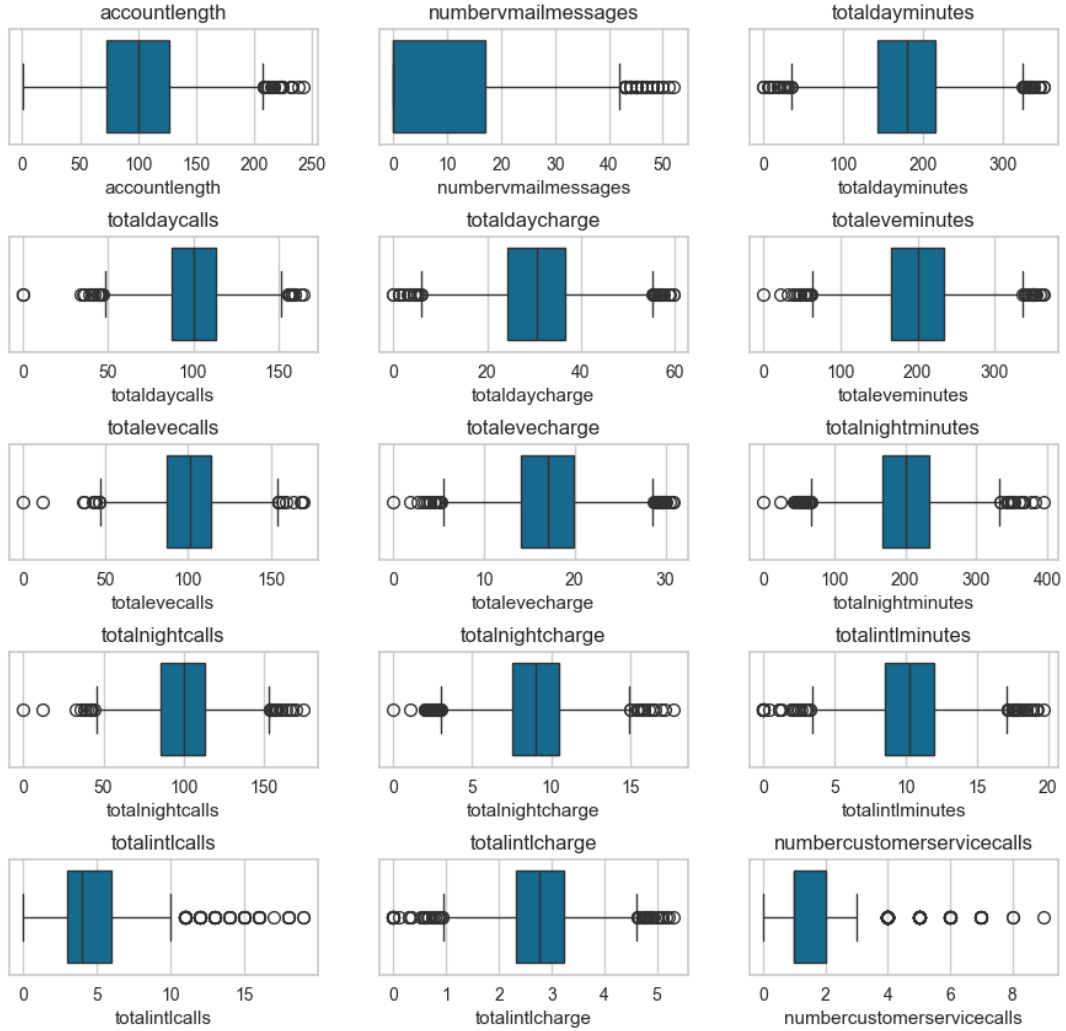
	accountlength	numbervmailmessages	totaldayminutes	totaldaycalls	totaldaycharge	totaleveminutes	totalevecalls
count	4951.000000	4950.000000	4950.000000	4950.000000	4950.000000	4950.000000	4950.000000
mean	100.238295	7.763636	180.306625	100.038788	30.629386	200.679798	100.243838
std	39.718817	13.552928	53.926625	19.844529	9.148881	50.486434	19.837380
min	1.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	73.000000	0.000000	143.700000	87.000000	24.430000	166.425000	87.000000
50%	100.000000	0.000000	180.100000	100.000000	30.600000	201.000000	101.000000
75%	127.000000	17.000000	216.200000	113.000000	36.720000	234.100000	114.000000
max	243.000000	52.000000	351.500000	165.000000	59.760000	363.700000	170.000000

2. Data Understanding

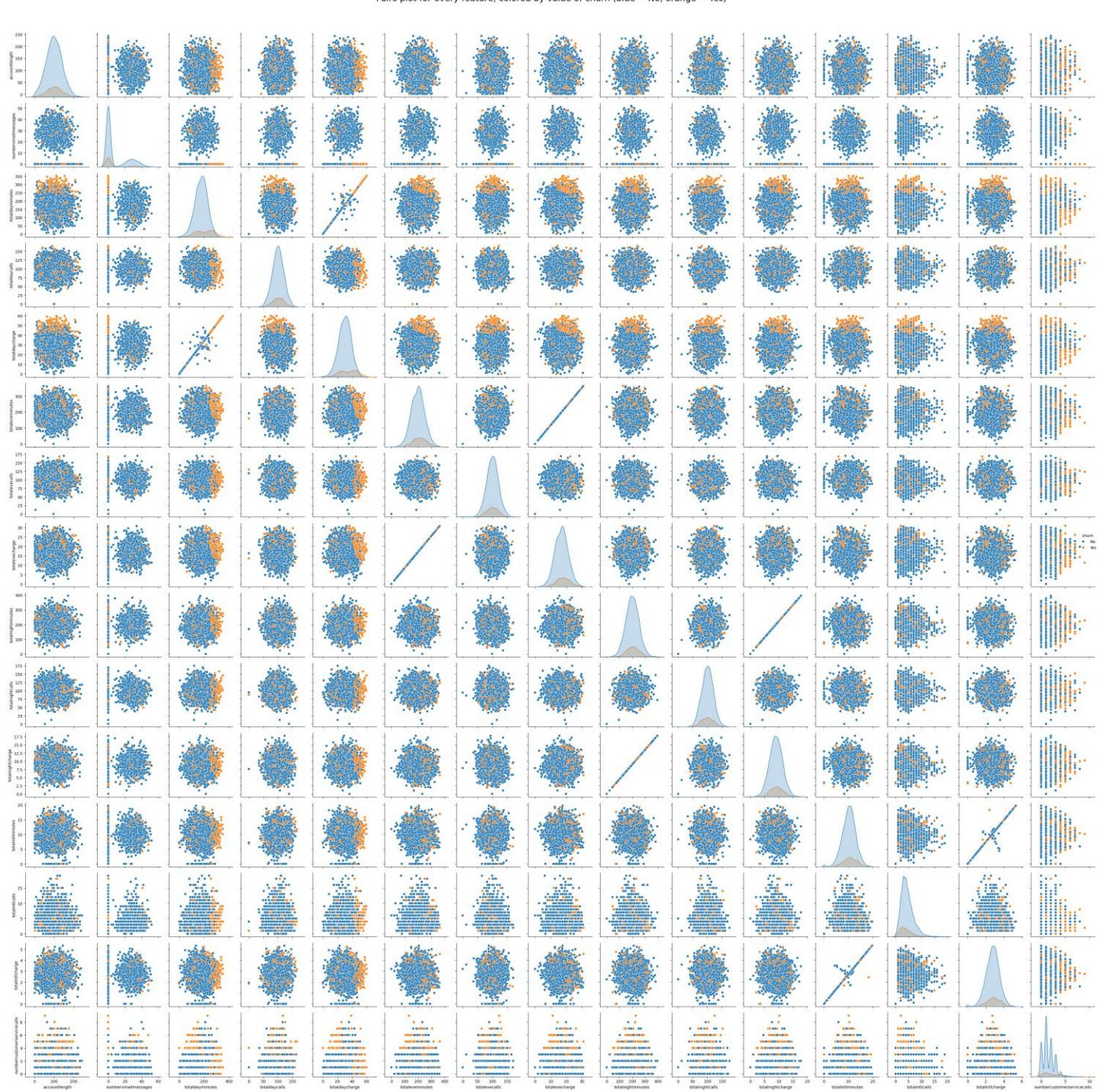
Histograms for selected features except numbervmailmessages



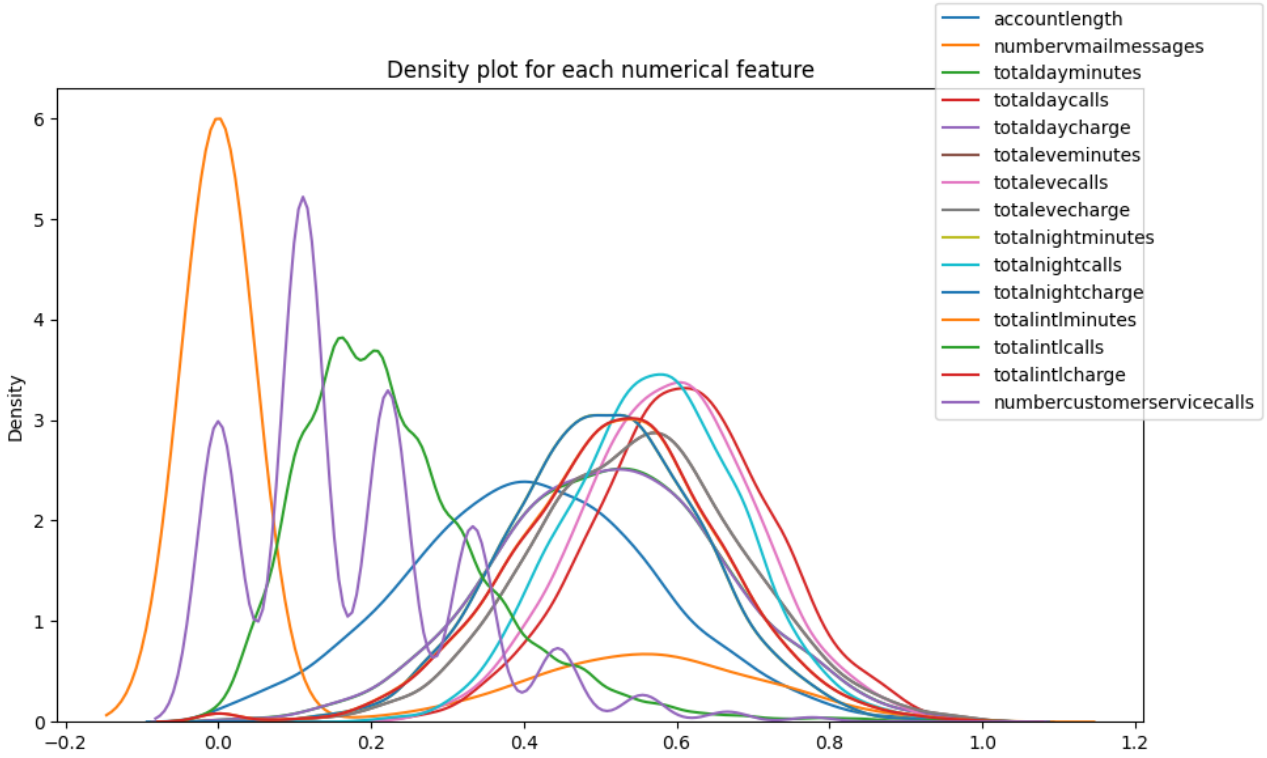
Boxplots for numeric features

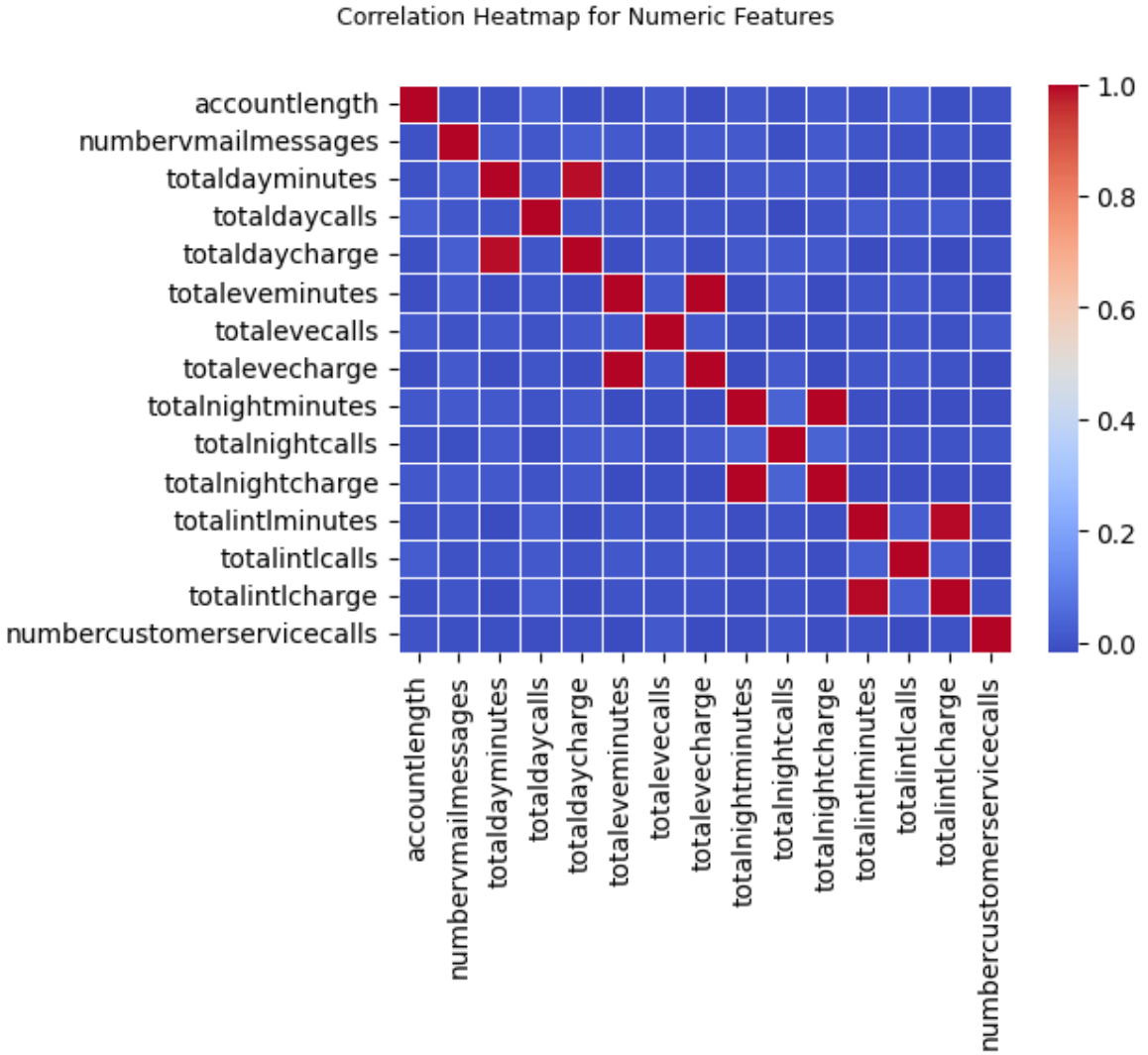


2. Data Understanding

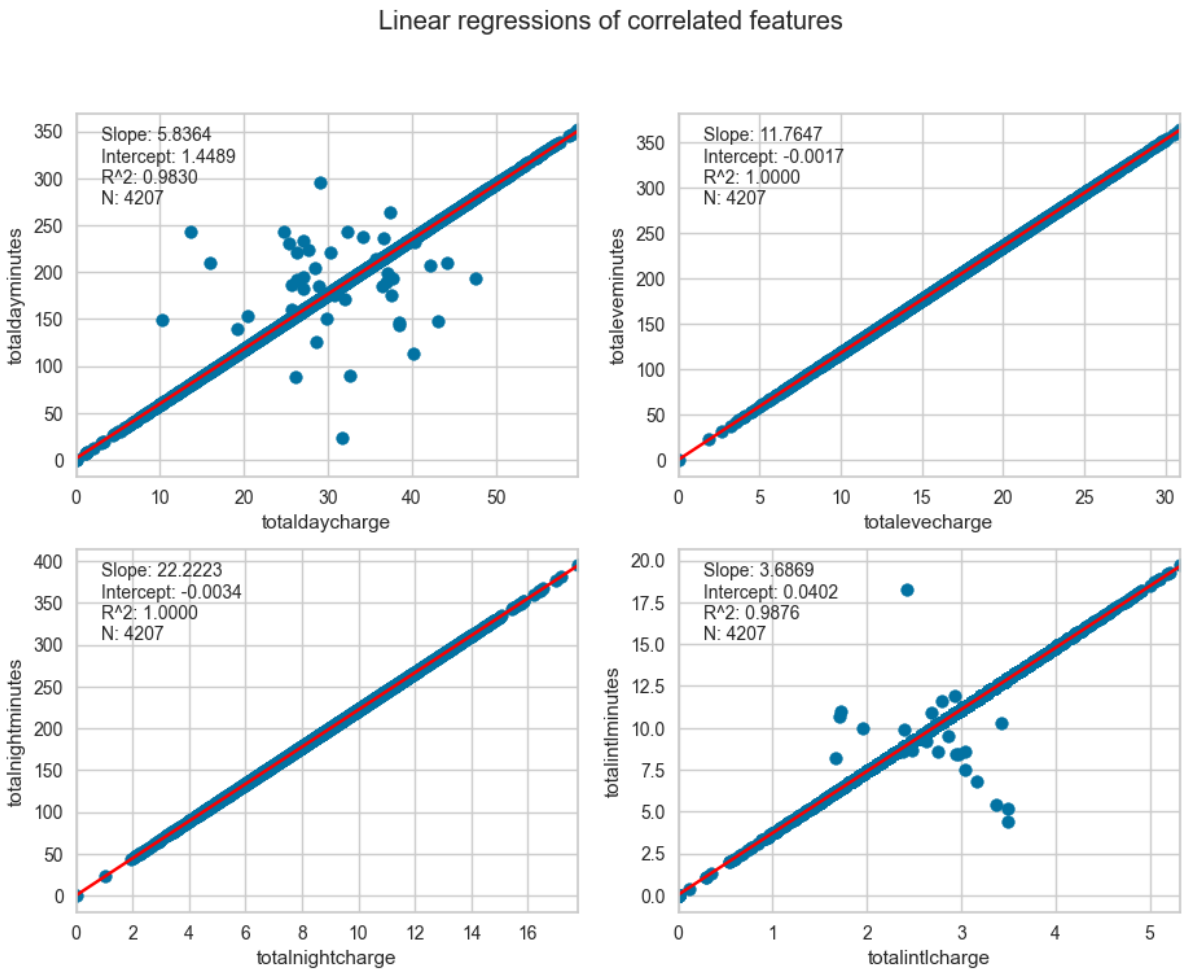


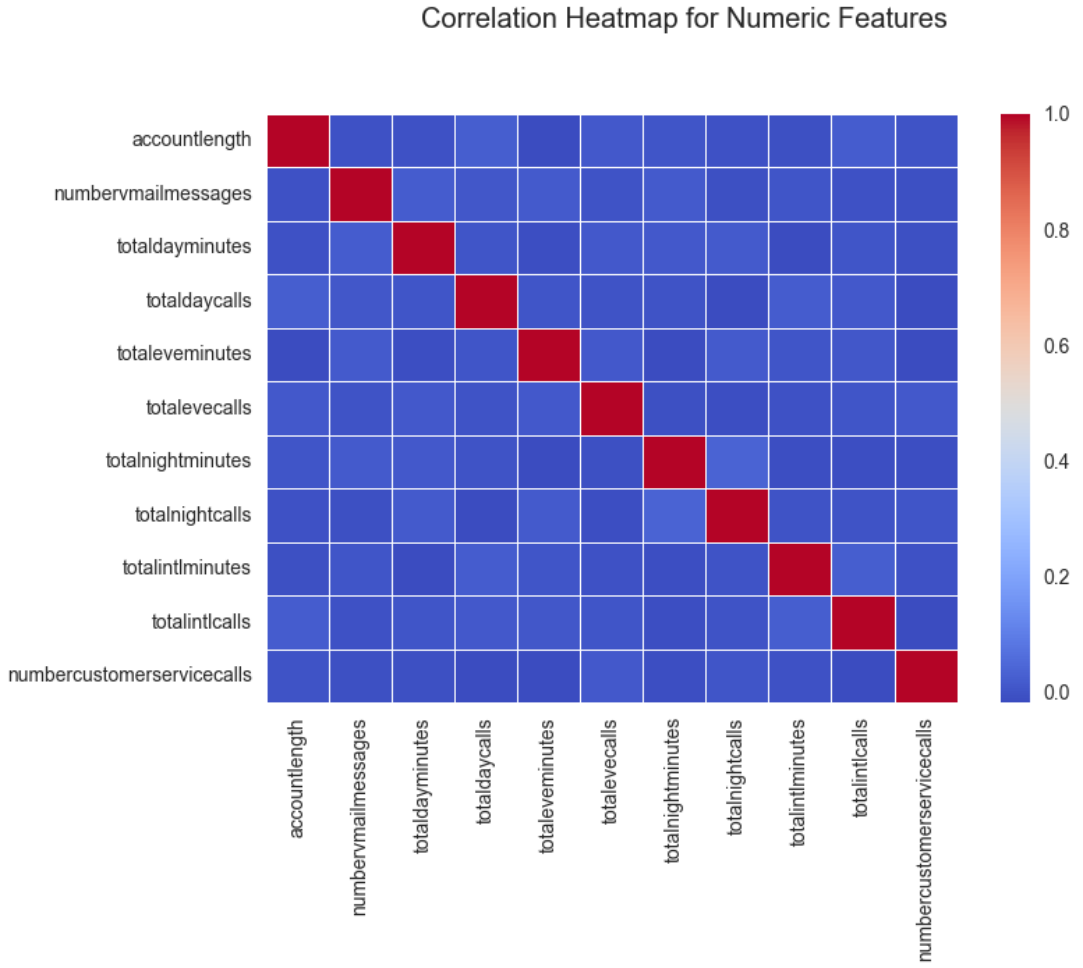
2. Data Understanding





2. Data Understanding





3. Data Preparation

3.1. Scaling

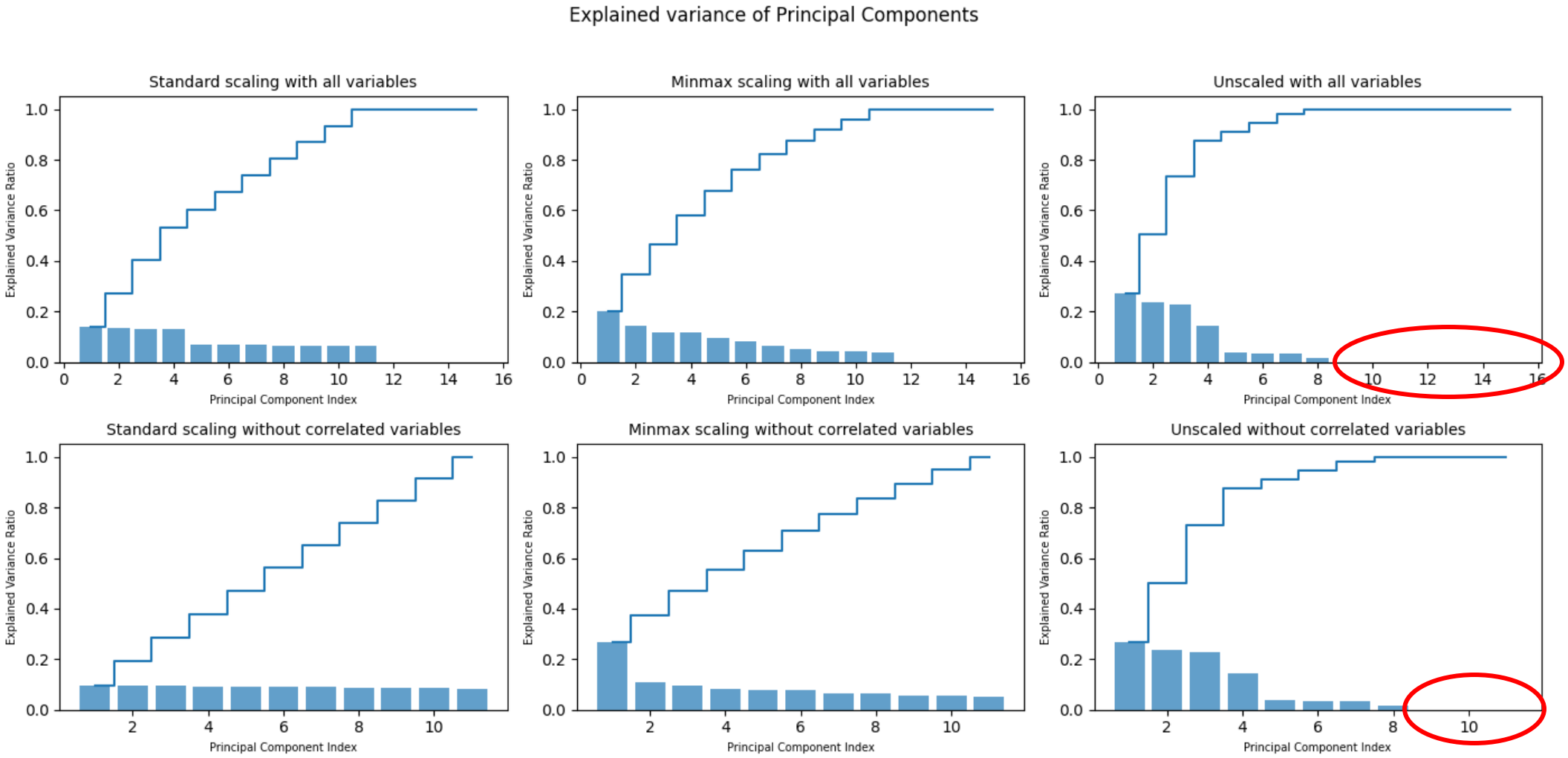
Defined a function **“scale”** that enables the user to:

- Choose between the standard and the min-max scaling (or no scaling at all);
- Drop or not the correlated columns.

3.2. Principal Component Analysis (PCA)

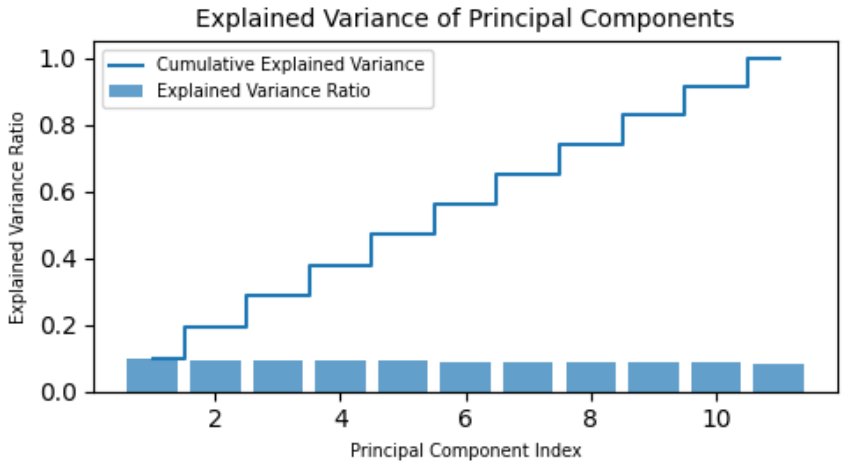
- Technique that **reduces** the high dimensional space of variables in data set;
- Finds an orthogonal basis where the data can be efficiently expressed, by linearly transforming the original data;

3. Data Preparation

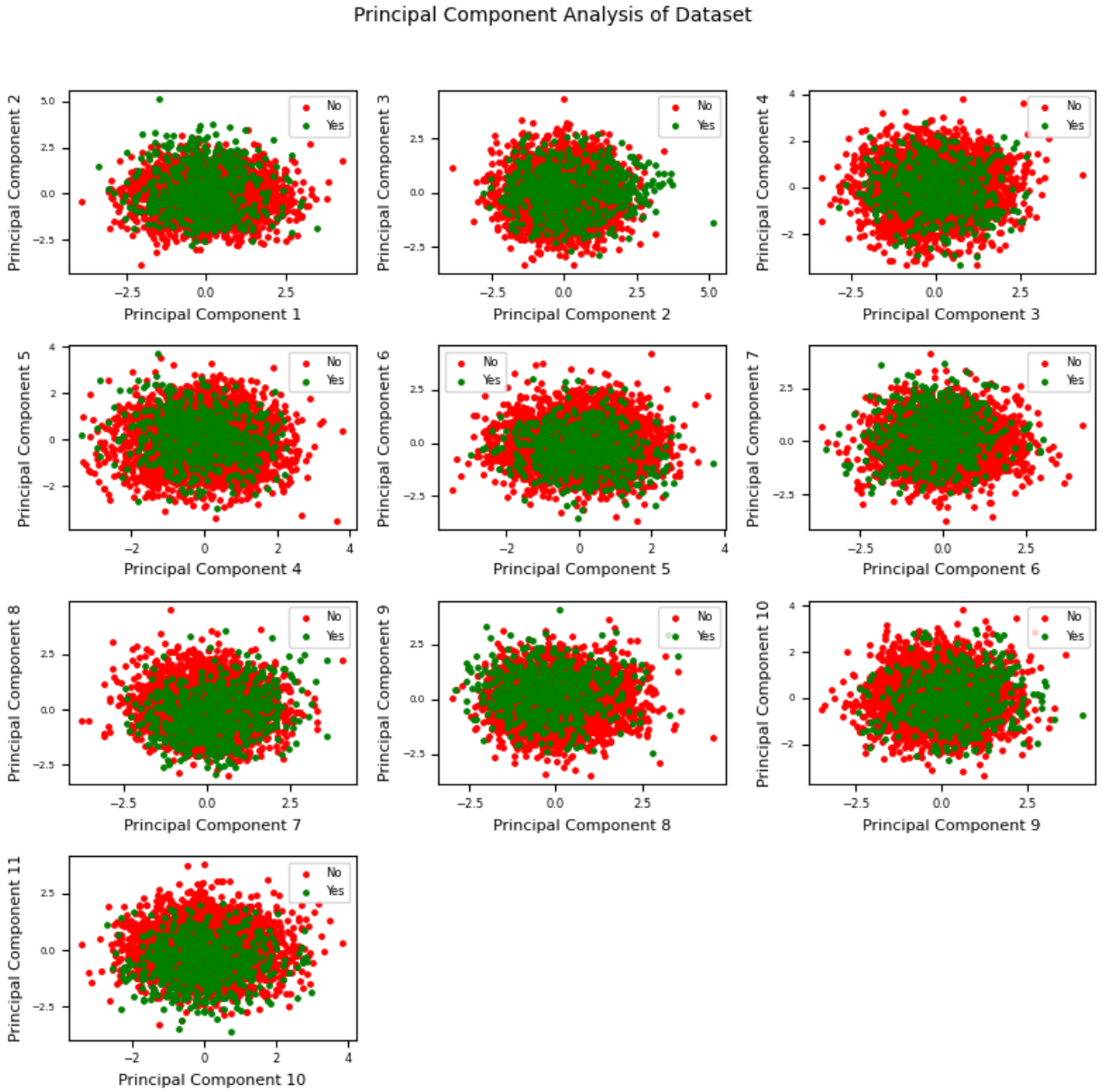


3. Data Preparation

Case for "standard" scaling without correlated variables:



- **PCA isn't very useful:**
  - performance of the classification models **did not improve significantly** (in some cases decreased)
  - **no defined** clustering visible
  - **cannot extract information** about the specifics of each variable (each one is turned to a linear combination of all of them)



4. Modelling

4.1. LDA and QDA

LDA and QDA: type of classification tool based on linear or quadratic decision boundaries.

Two main scenarios:

- 5-fold Cross-Validation
- Simple hold-out method, 80% train and 20% test

6 sub-scenarios, combinations of:

- Correlated variables are dropped or not
- Scaling the data: no scale, "minmax" or "standard" methods

4.1. LDA and QDA

Performance metrics for LDA with no scaling and lsqr / eigen solver.

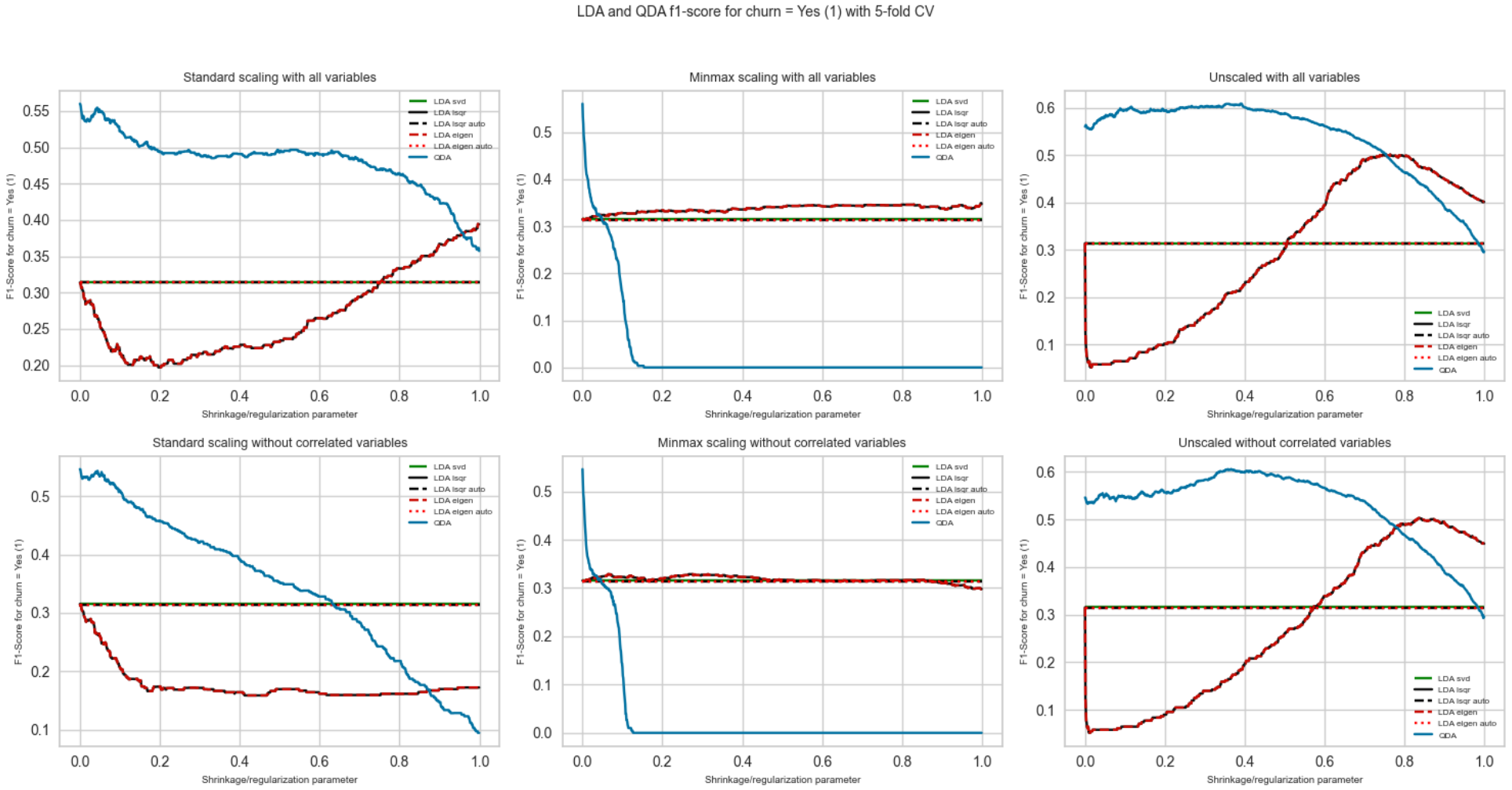
Correlated Variables	CV	Accuracy	F1-score	ROC AUC	Precision	Shrink_param
Yes	No	0.898	0.543	0.700	0.888	0.870
No	No	0.897	0.545	0.702	0.887	0.806
Yes	Yes	0.898	0.527	0.689	0.889	0.752
No	Yes	0.898	0.527	0.689	0.889	0.835

Performance metrics for QDA with no scaling.

Correlated Variables	CV	Accuracy	F1-score	ROC AUC	Precision	Reg_param
Yes	No	0.890	0.597	0.757	0.886	0.049
No	No	0.897	0.592	0.740	0.888	0.042
Yes	Yes	0.885	0.580	0.747	0.881	0.389
No	Yes	0.894	0.578	0.732	0.885	0.365

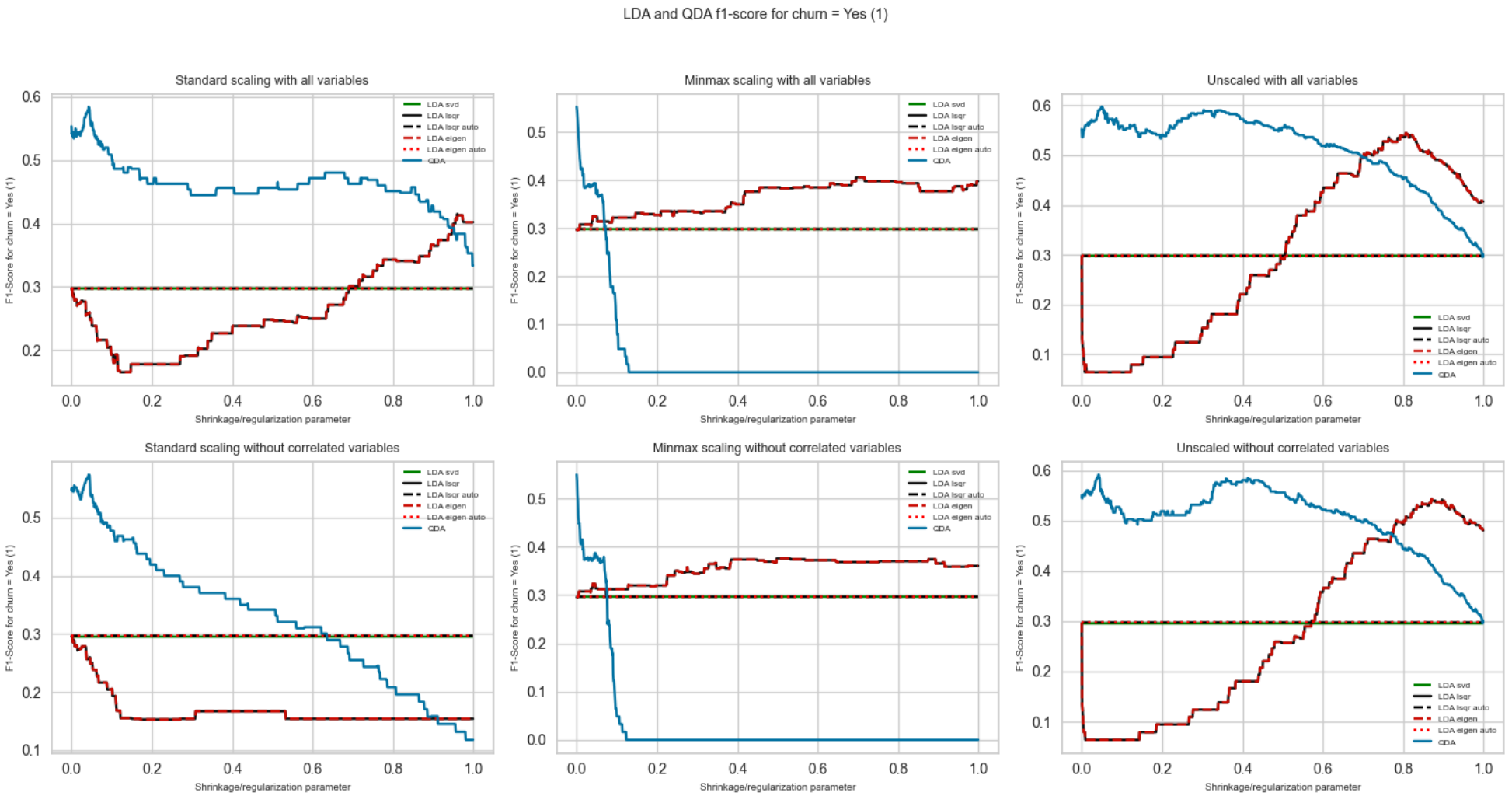
4.1. LDA and QDA

F1-Score for churn = Yes (1) with CV



4.1. LDA and QDA

F1-Score for churn = Yes (1) with holdout



4.2. Multi-layer Perceptron

- **Perceptron**: simplest and most fundamental NN unit

- Side by side preceptrons: **multilayer perceptron**.

- **5-fold cross-validation**, as it was a user friendly option.

- Immense computing time -> restricted possible values for model parameters by trial and error:

  - Four activation types: identity, logistic, tanh and relu.
  - Three solvers: lbfgs, sgd and adam.
  - Six hidden layer sizes: (9,),(10,),(11,),(100,),(10,5,) and (100,50).
  - Nine alphas: 0, 0.001, 0.005, 0.01, 0.05, 0.1, 0.2, 0.5 and 0.9.

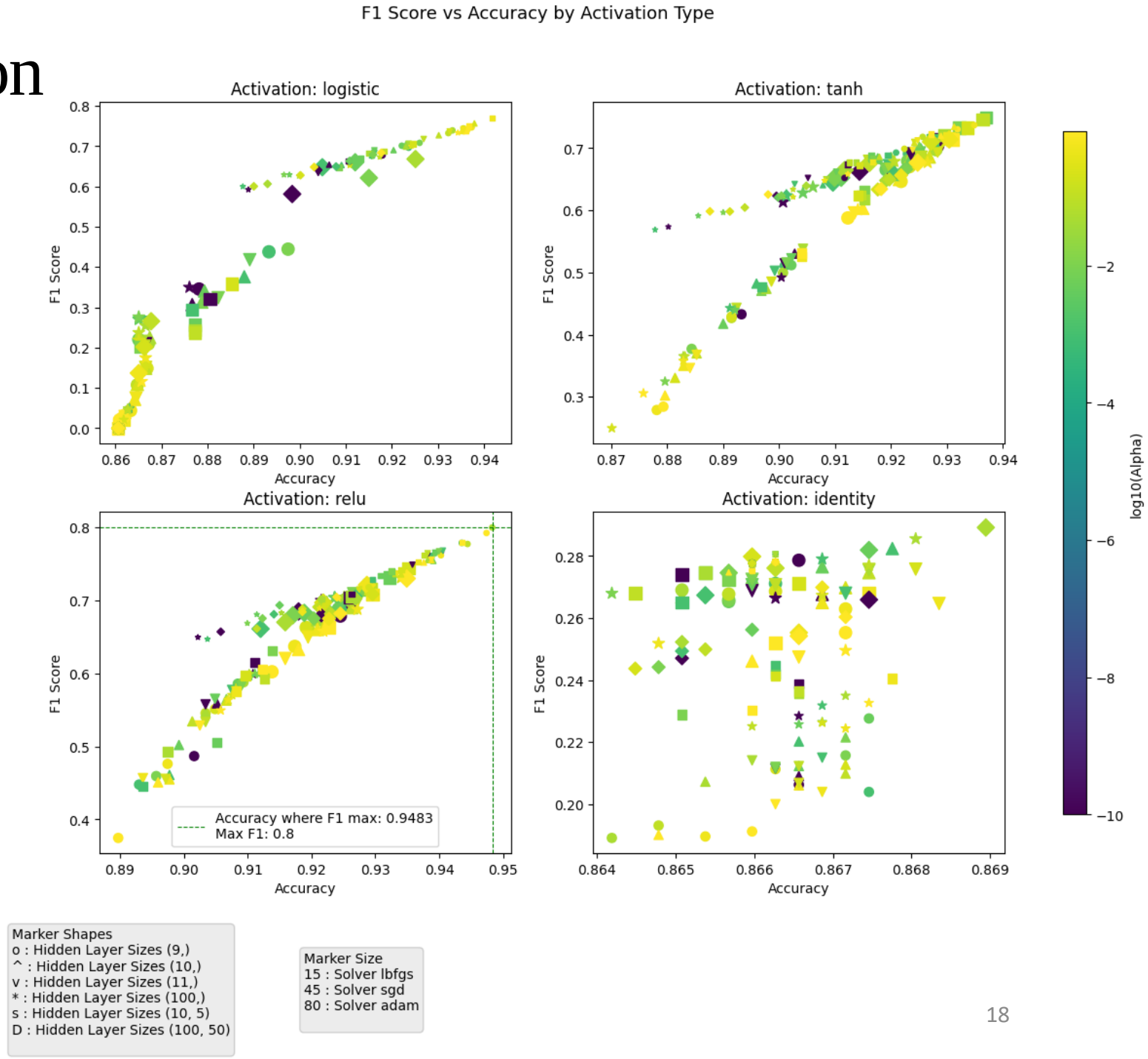
- We used **standard scaling** and **dropped** correlated variables. Using the raw data resulted in a poor performance.

4.2. Multi-layer Perceptron

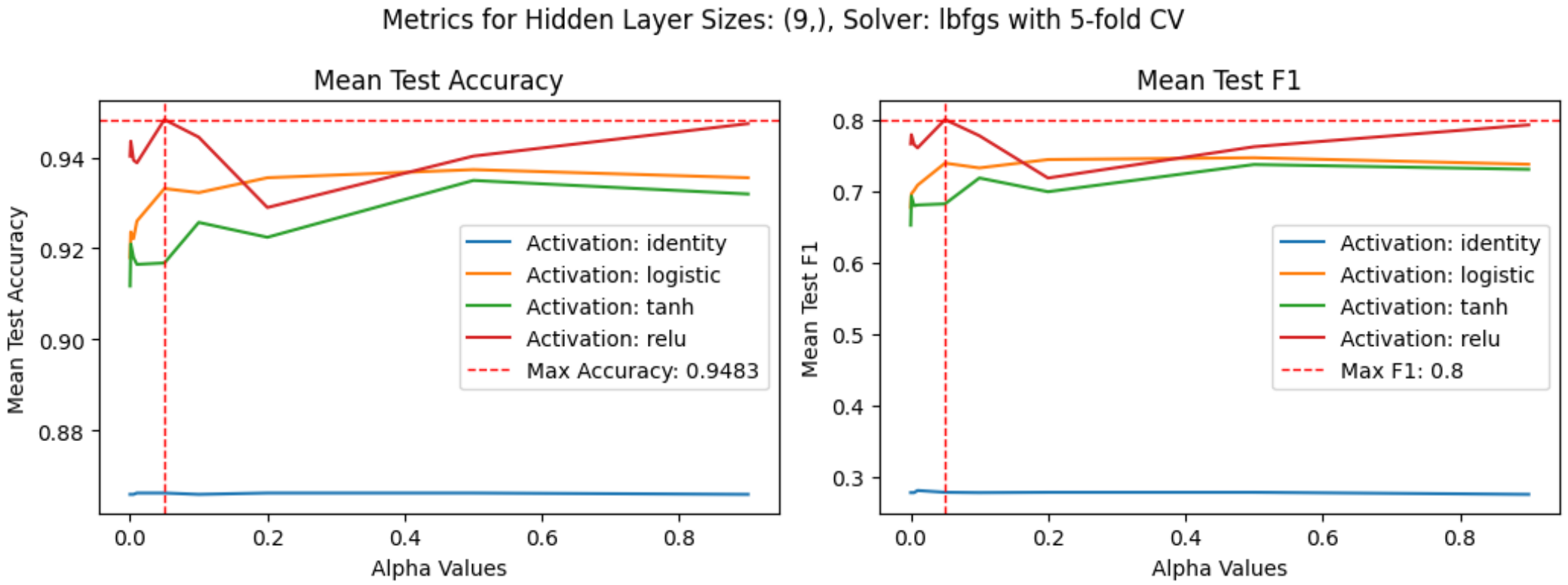
Best model: MLPClassifier with

- Activation = relu
- Alpha = 0.05;
- Hidden layer size = (9,)
- Solver = lbfgs

Note: solvers sgd (stochastic gradient descent) and adam tend to have poorer performances!

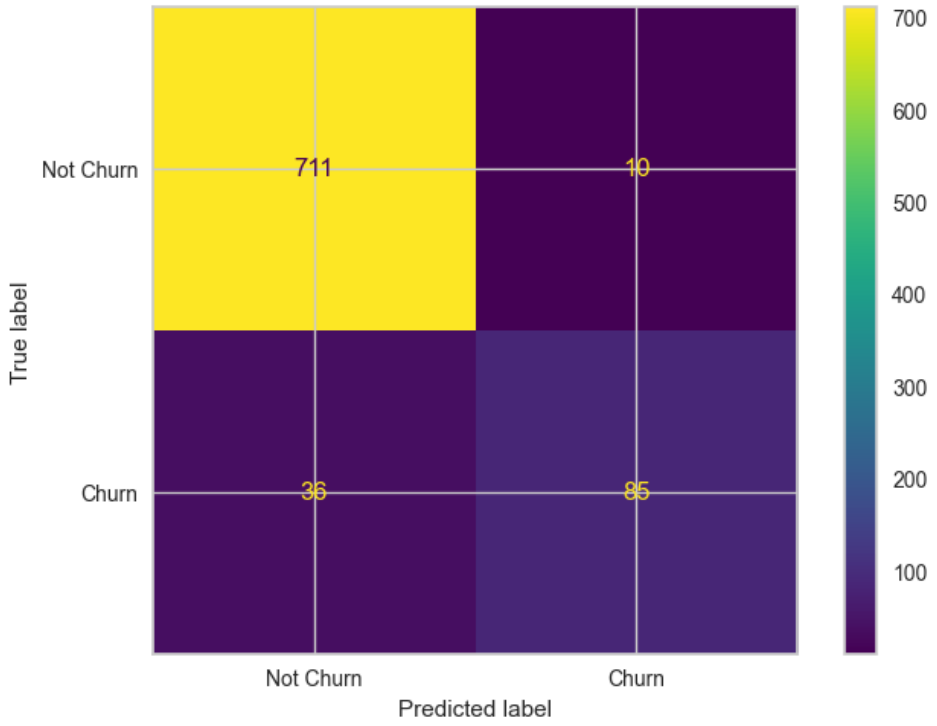


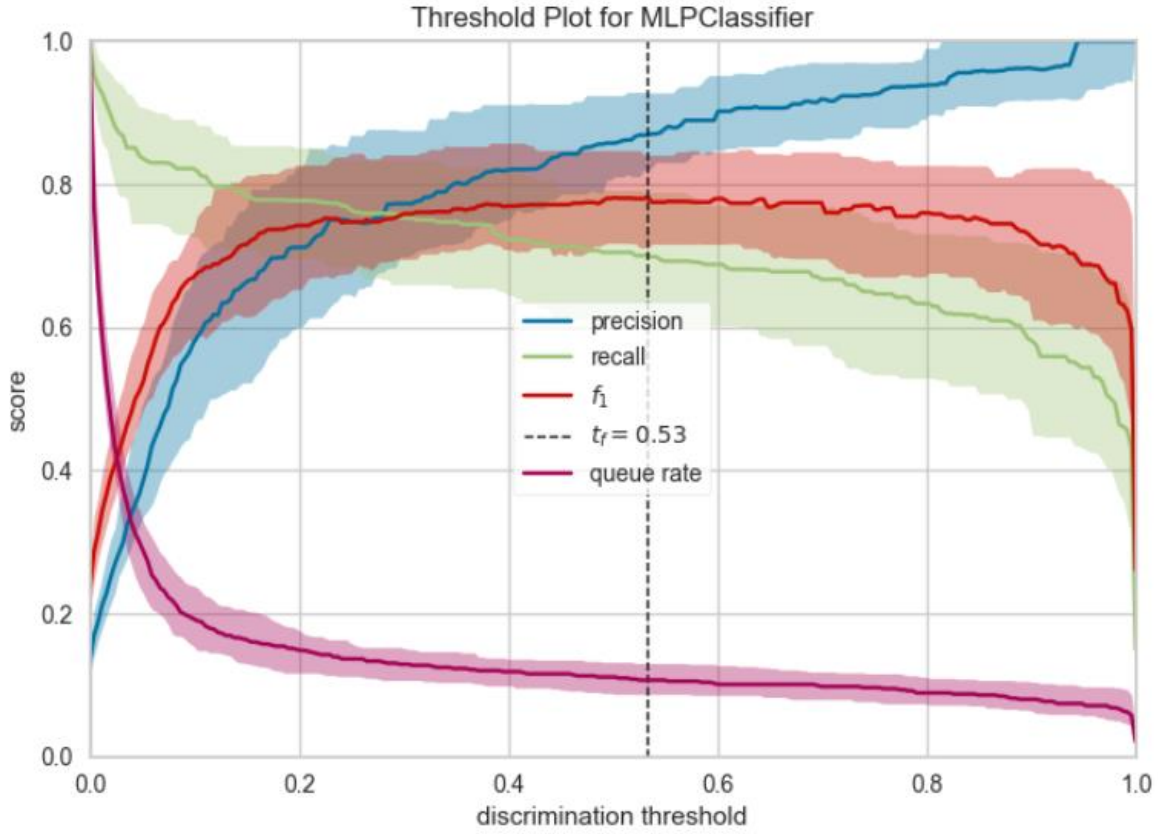
4.2. Multi-layer Perceptron



4.2. Multi-layer Perceptron

Confusion Matrix for the best model:





4.3. Bayesian Classifier

Two main scenarios:

- 10-fold Cross-Validation
- Simple hold-out method, 80% train and 20% test

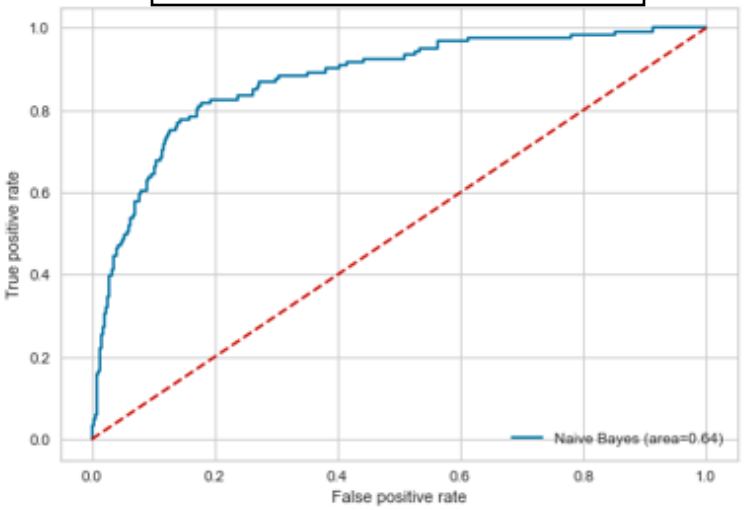
In each scenario, we considered **4 sub-scenarios**, combinations of:

- Correlated variables are dropped or not
- Scaling the data: "minmax" or "standard" methods

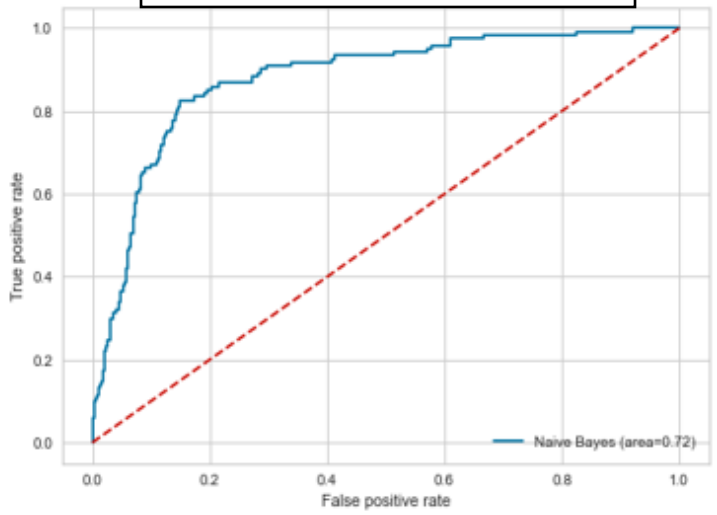
Scaling Type	Correlated Variables	Cross-Validation	Accuracy	F1-score	Precision	Roc_auc
Standard	Yes	No	0.869	0.526	0.550	0.72
Standard	No	No	0.857	0.455	0.505	0.67
Standard	Yes	Yes	0.882	0.428	0.712	0.64
Standard	No	Yes	0.869	0.295	0.657	0.59
Minmax	Yes	No	0.869	0.526	0.550	0.72
Minmax	No	No	0.857	0.455	0.505	0.67
Minmax	Yes	Yes	0.869	0.526	0.550	0.72
Minmax	No	Yes	0.857	0.455	0.505	0.67

4.3. Bayesian Classifier

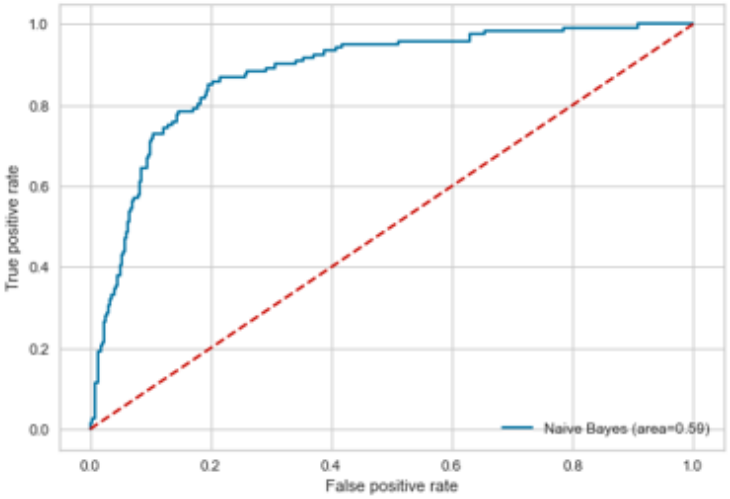
Standard scaling with all variables



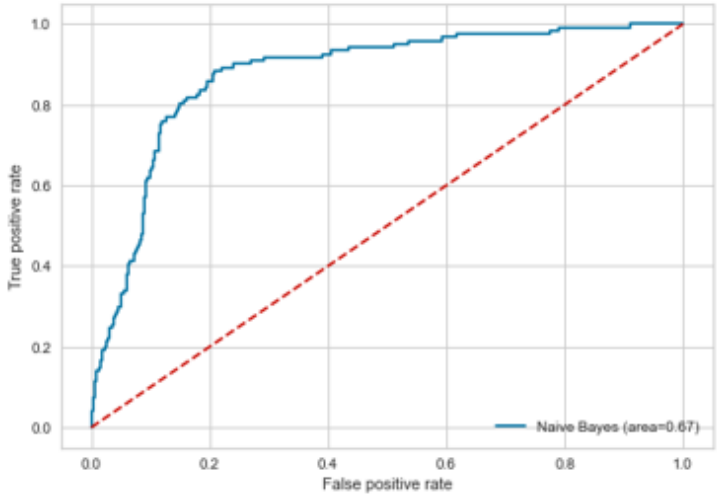
Minmax scaling with all variables



Standard scaling without correlated variables



Minmax scaling without correlated variables



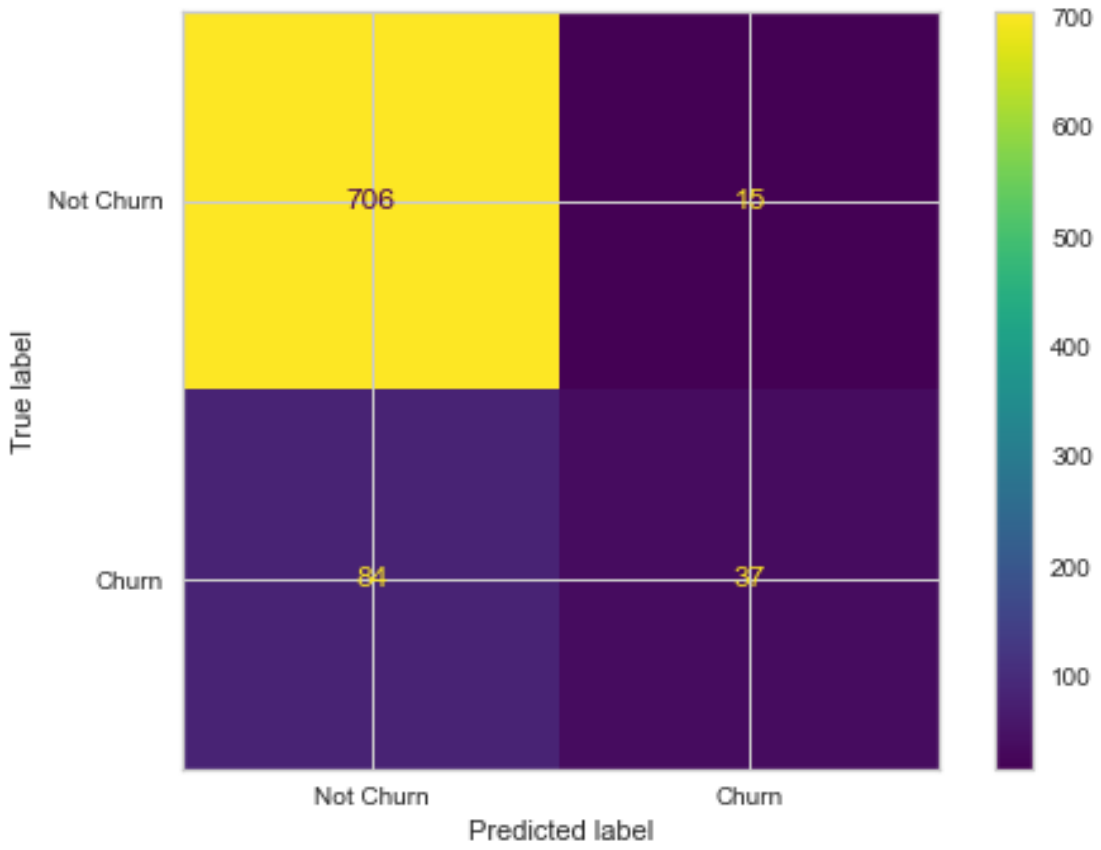
Best hyperparameters:
priors: None
var_smoothing: 0.1

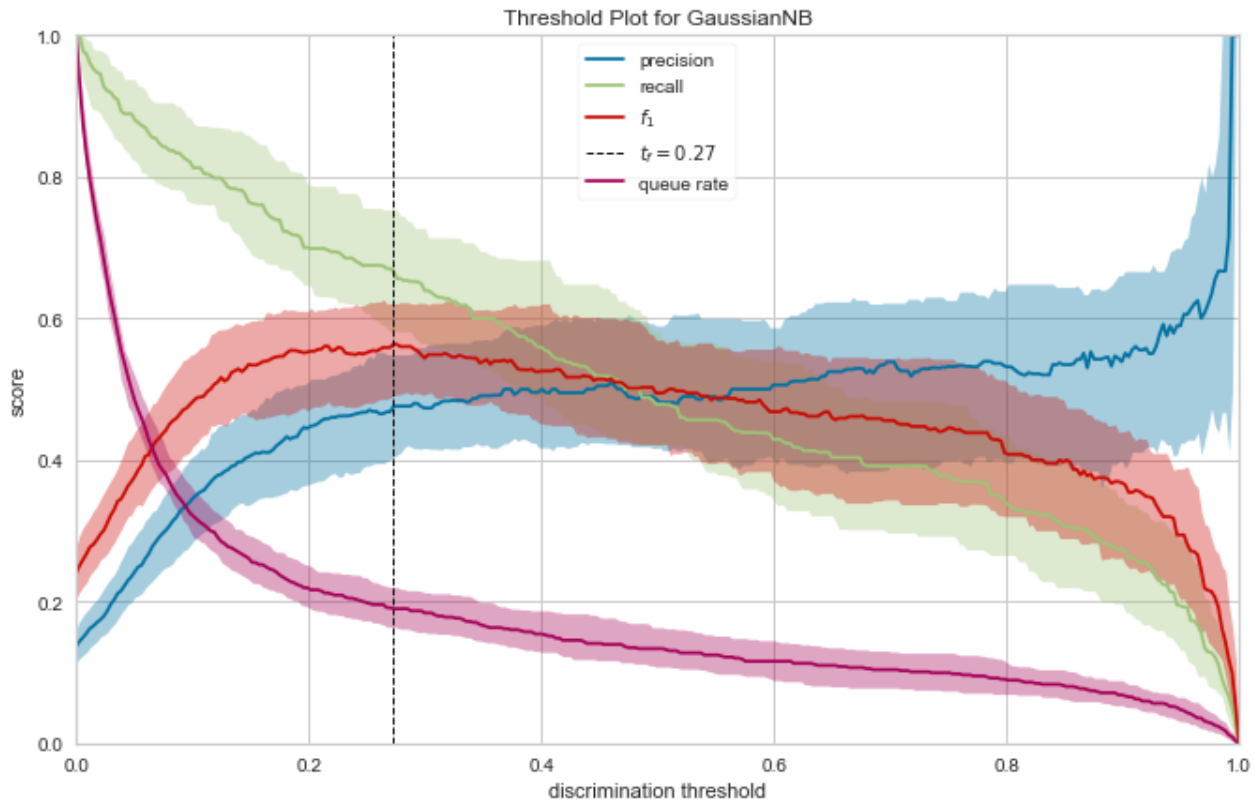
ROC curves for the 4 cases **obtained with cross-validation**.

- Higher AUC→ Better predictive accuracy
- The case with higher AUC corresponds to the minmax scaling with all variables, but as seen before the best model is the one with standard scaling and all variables.

4.3. Bayesian Classifier

Confusion matrix and **threshold plot** for the train dataset using the best model with **standard scaling, all variables** and with **cross-validation**.





4.4. Logistic Classification

Two main scenarios:

- 10-fold Cross-Validation
- Simple hold-out method, 80% train and 20% test

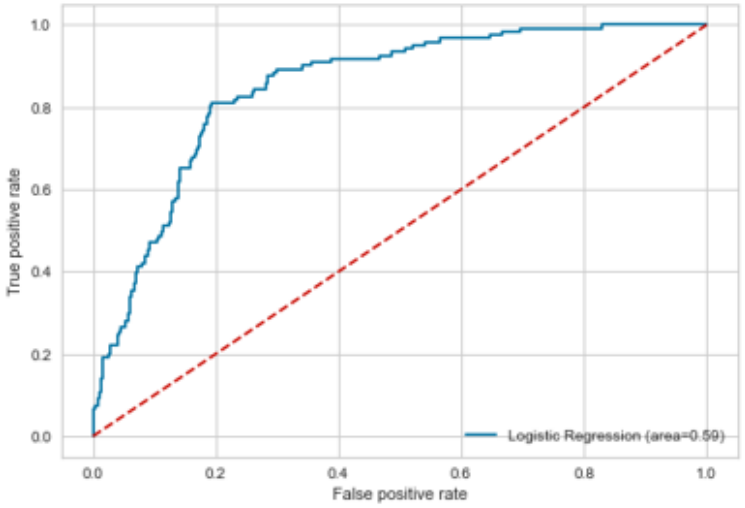
In each scenario, we considered **4 sub-scenarios**, combinations of:

- Correlated variables are dropped or not
- Scaling the data: "minmax" or "standard" methods

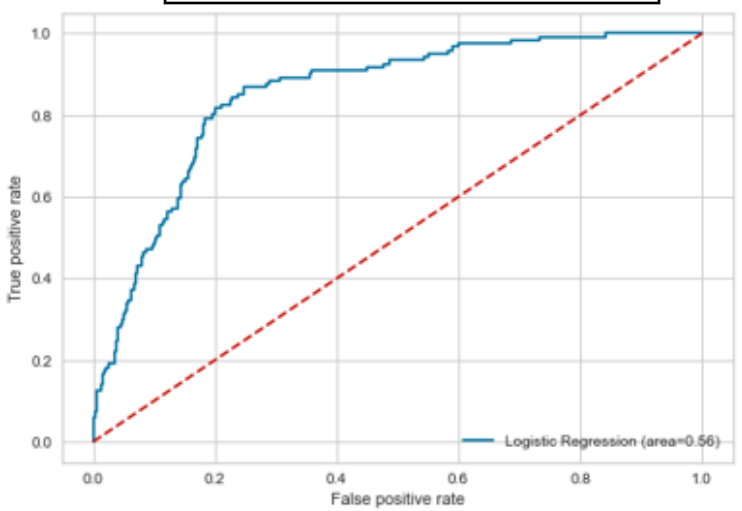
Scaling Type	Correlated Variables	Cross-Validation	Accuracy	F1-score	Precision	Roc_auc
Standard	Yes	No	0.869	0.295	0.657	0.59
Standard	No	No	0.869	0.295	0.657	0.59
Standard	Yes	Yes	0.871	0.297	0.676	0.59
Standard	No	Yes	0.862	0.216	0.593	0.56
Minmax	Yes	No	0.867	0.263	0.645	0.58
Minmax	No	No	0.867	0.263	0.645	0.58
Minmax	Yes	Yes	0.865	0.230	0.630	0.56
Minmax	No	Yes	0.865	0.230	0.630	0.56

4.4. Logistic Classification

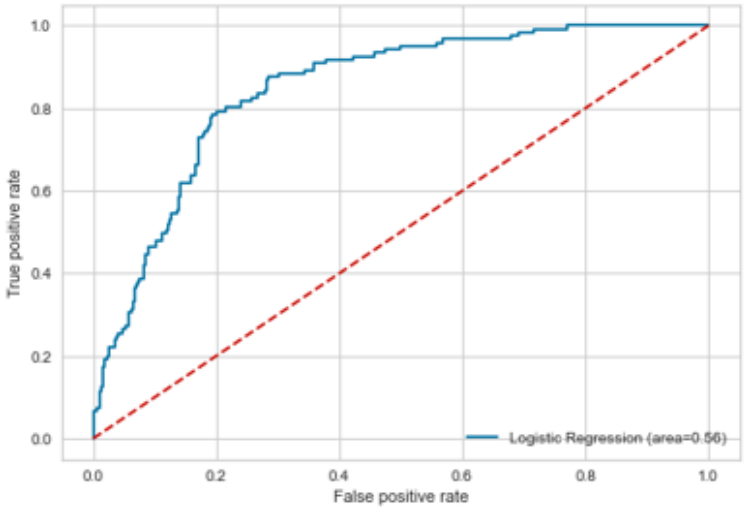
Standard scaling with all variables



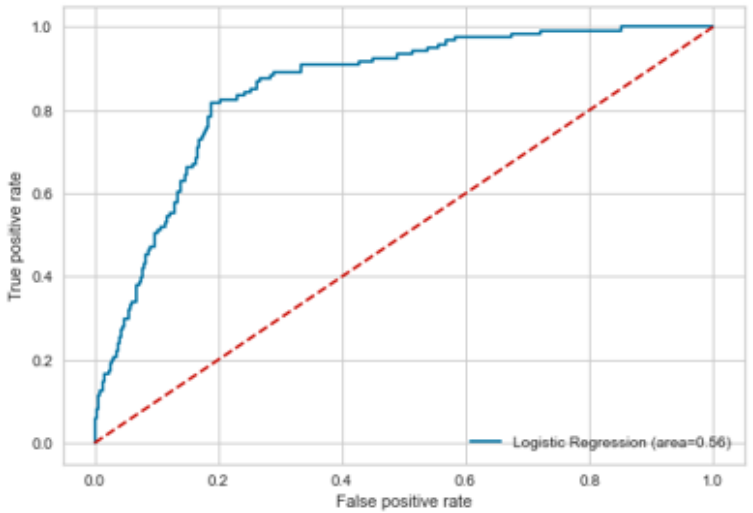
Minmax scaling with all variables



Standard scaling without correlated variables



Minmax scaling without correlated variables



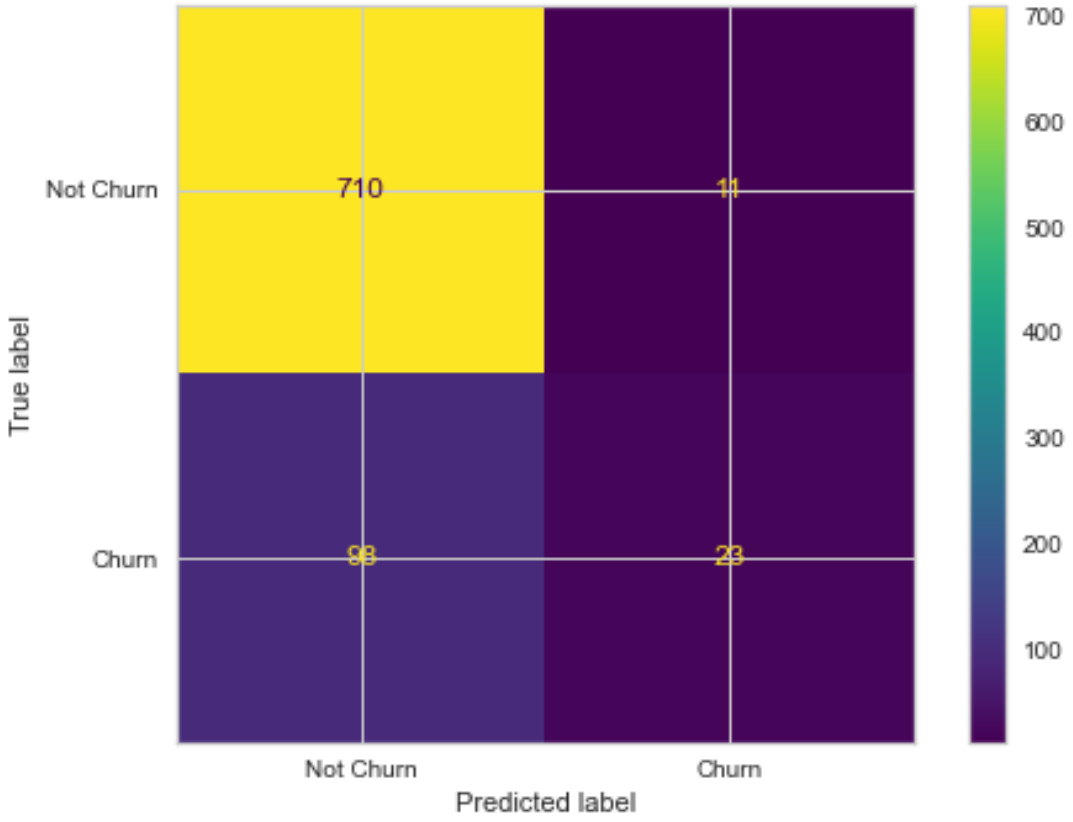
Best hyperparameters:
C: 0.5623413251903491
penalty: l2

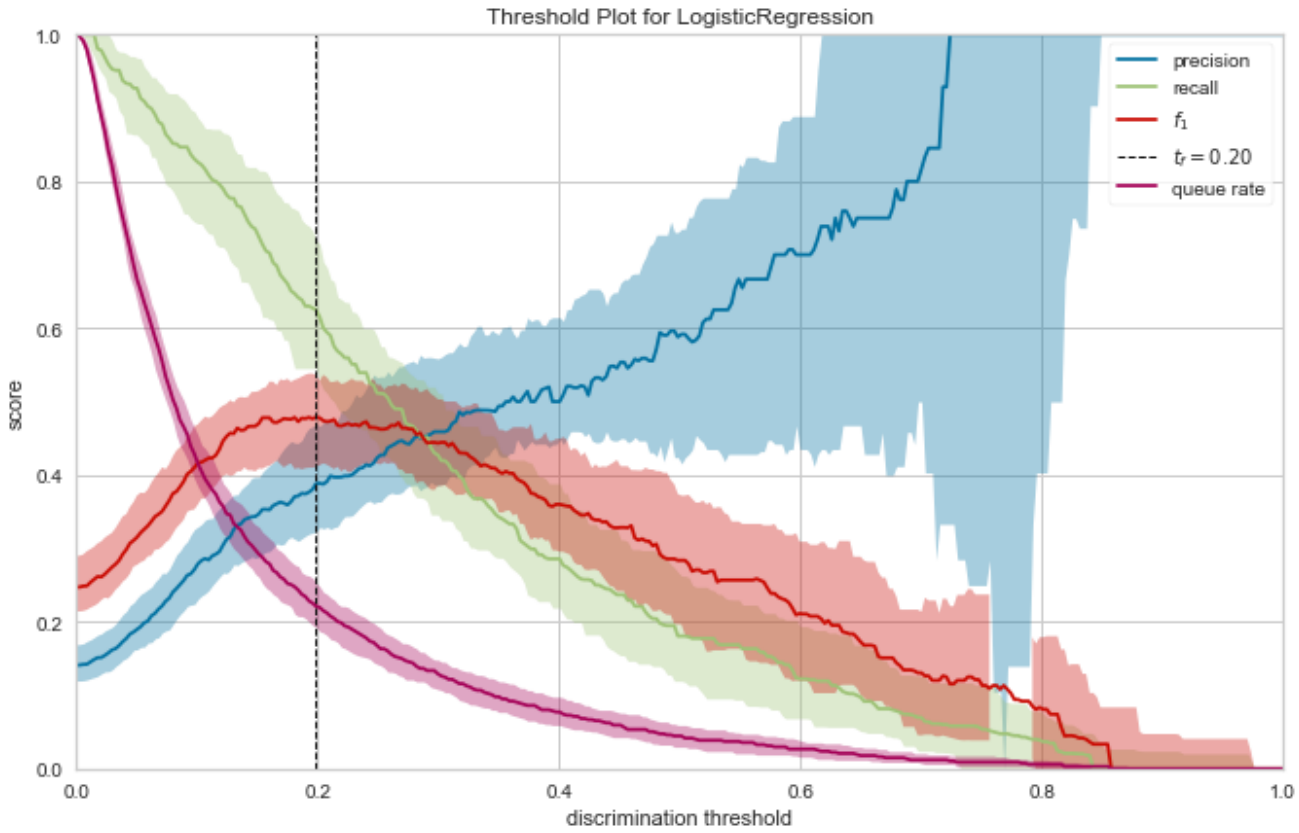
ROC curves for the 4 cases **obtained with cross-validation**.

- The case with higher AUC and also the best model is the one with standard scaling and all variables.

4.4. Logistic Classification

Confusion matrix and **threshold plot** for the train dataset using the best model with **standard scaling, all variables** and with **cross-validation**.





4.5. kNN Classifier

Two main scenarios:

- 10-fold Cross-Validation
- Simple hold-out method, 80% train and 20% test

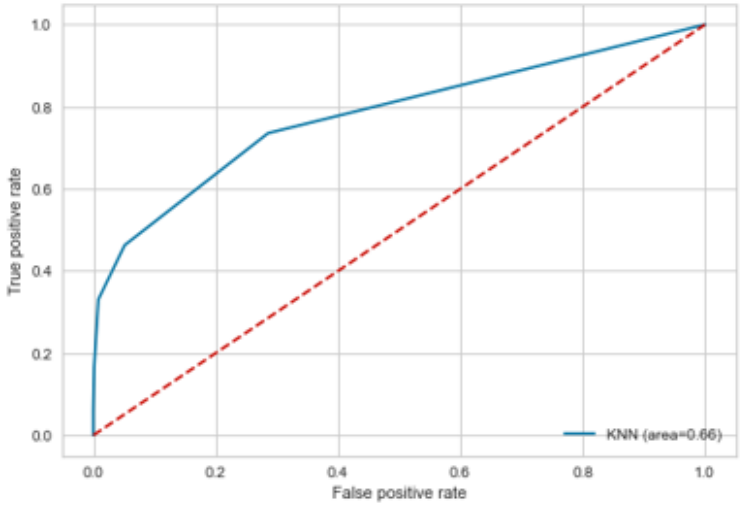
In each scenario, we considered **4 sub-scenarios**, combinations of:

- Correlated variables are dropped or not
- Scaling the data: "minmax" or "standard" methods

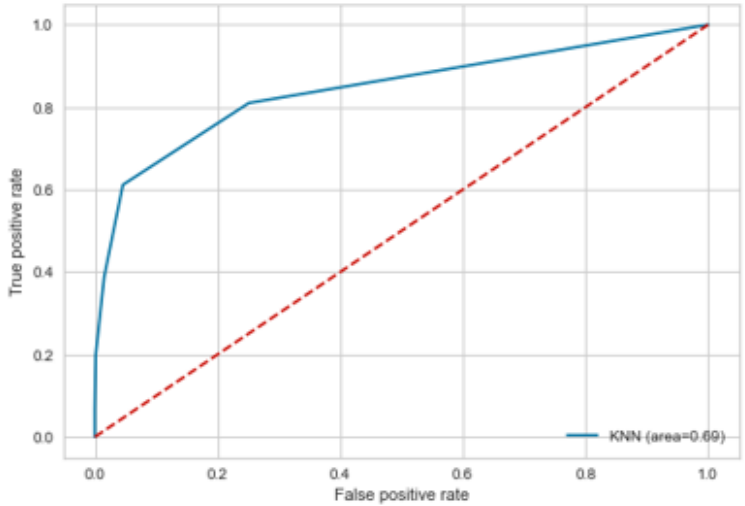
Scaling Type	Correlated Variables	Cross-Validation	Accuracy	F1-score	Precision	Roc_auc
Standard	Yes	No	0.886	0.351	0.963	0.61
Standard	No	No	0.874	0.243	0.895	0.57
Standard	Yes	Yes	0.897	0.479	0.870	0.66
Standard	No	Yes	0.888	0.420	0.829	0.64
Minmax	Yes	No	0.898	0.456	0.973	0.65
Minmax	No	No	0.882	0.344	0.867	0.60
Minmax	Yes	Yes	0.899	0.525	0.810	0.69
Minmax	No	Yes	0.904	0.537	0.870	0.69

4.5. kNN Classifier

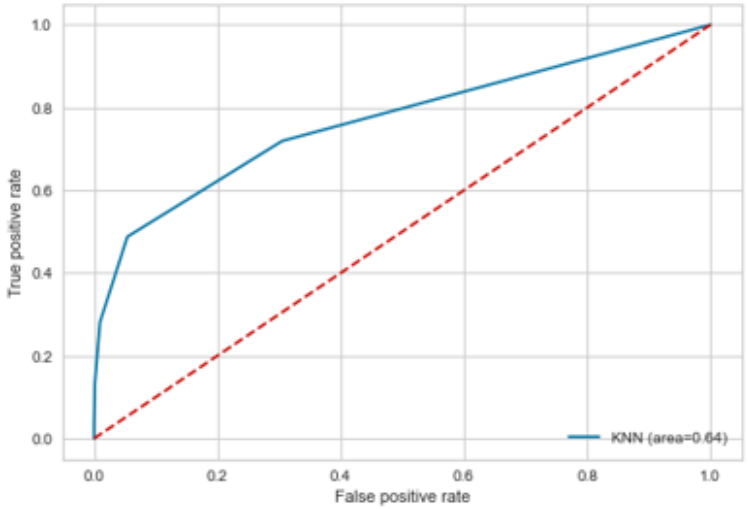
Standard scaling with all variables



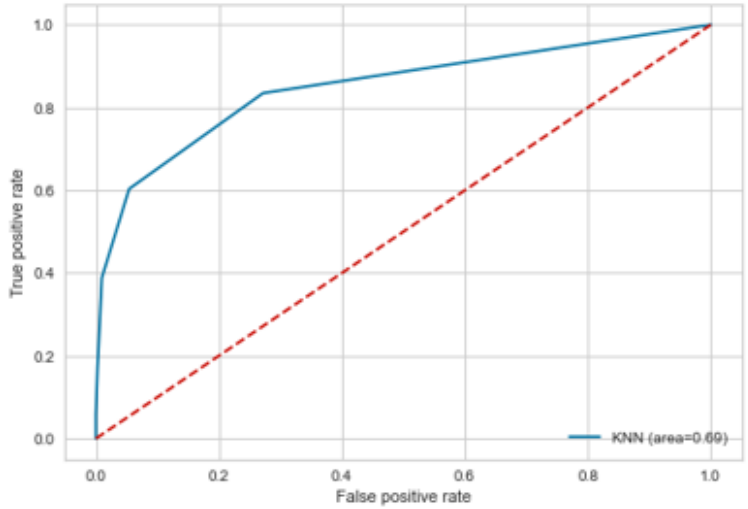
Minmax scaling with all variables



Standard scaling without correlated variables



Minmax scaling without correlated variables



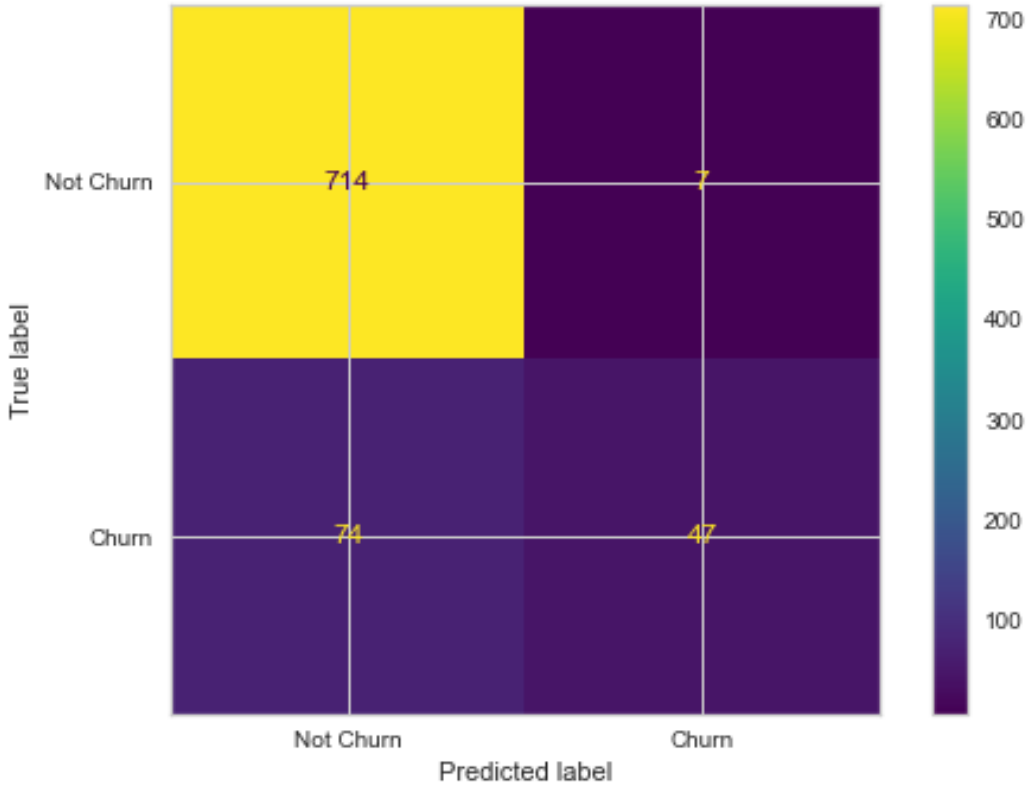
Best hyperparameters:
n_neighbours: 5

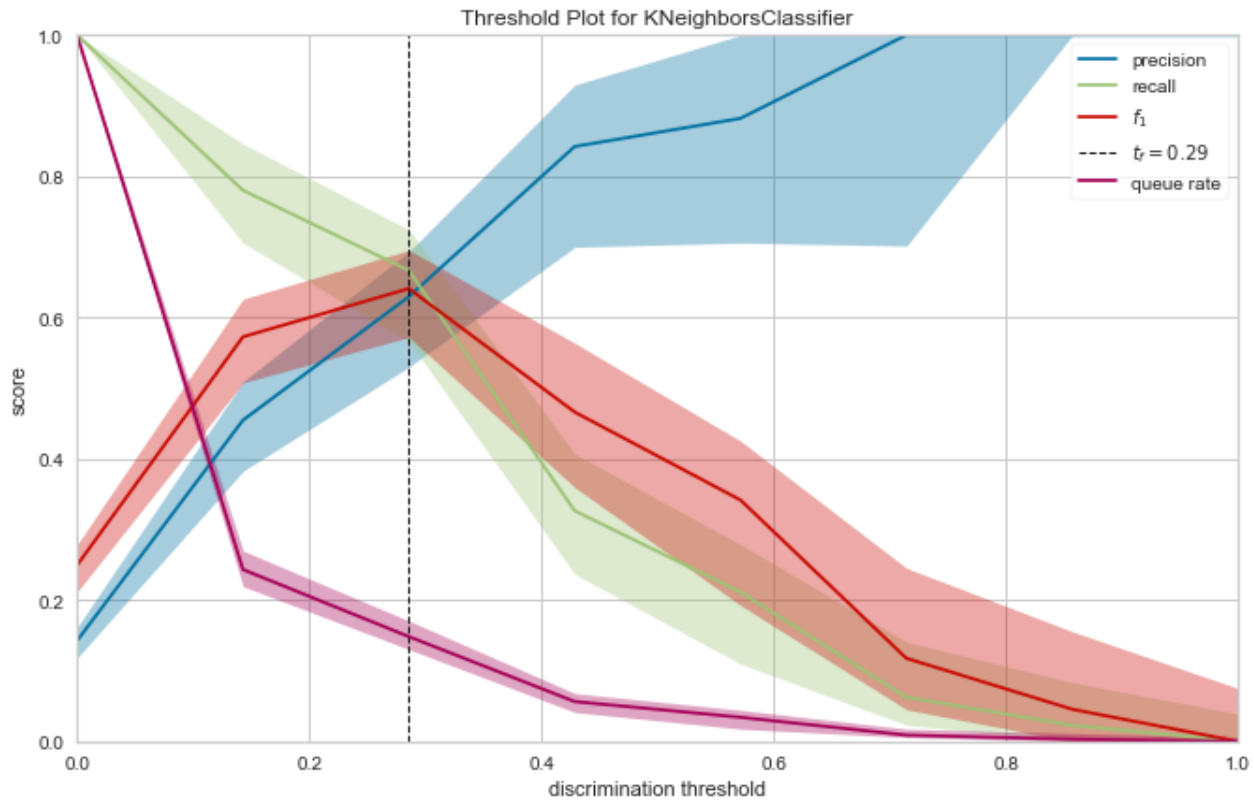
ROC curves for the 4 cases **obtained with cross-validation**.

- The best model is the one with minmax scaling and without the correlated variables.

4.5. kNN Classifier

Confusion matrix and **threshold plot** for the train dataset using the best model with **minmax scaling, without the correlated variables** and with **cross-validation**.





4.6. Decision Trees

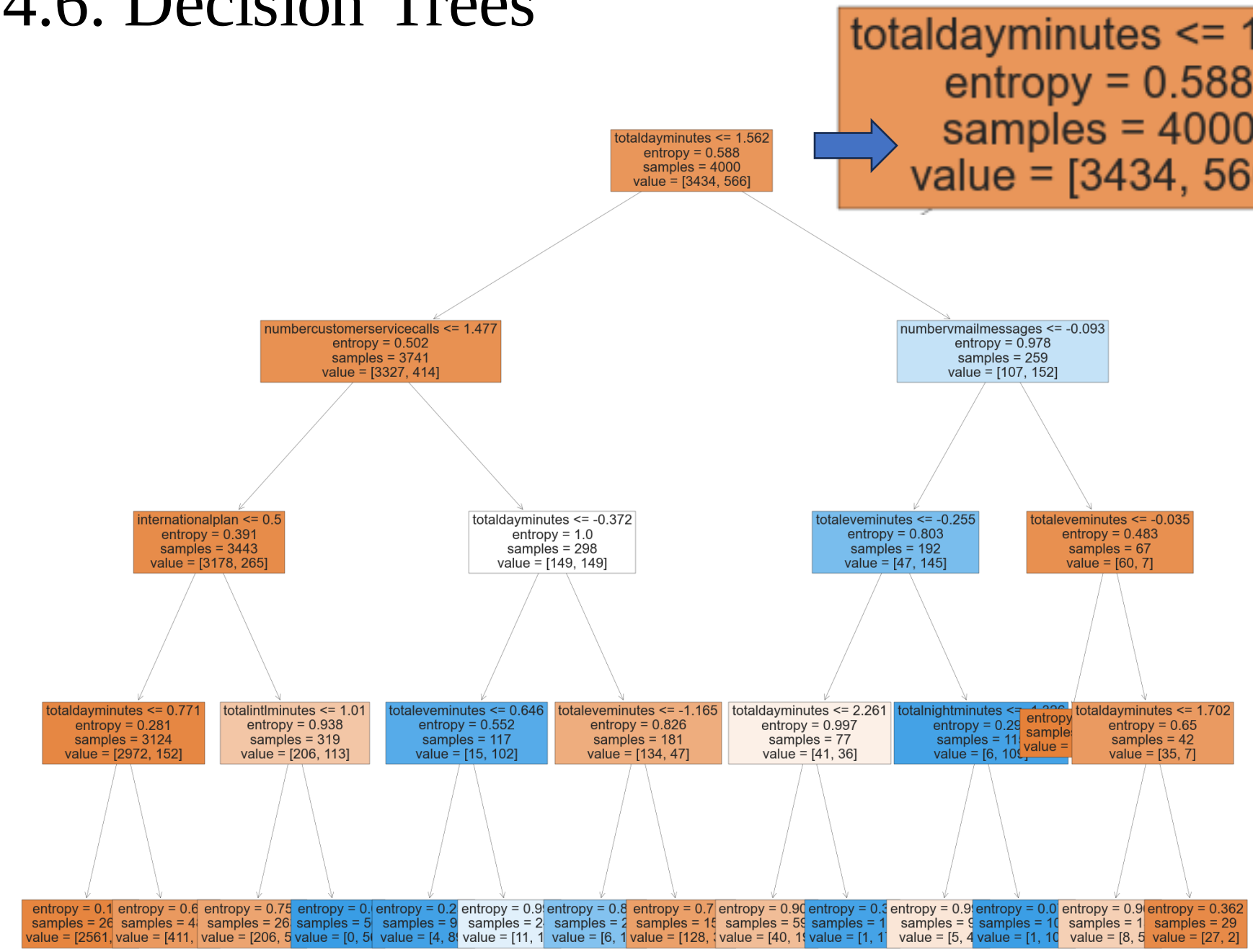
	param_criterion	param_max_depth	param_min_samples_leaf	param_min_samples_split	param_splitter	mean_test_score
3320	entropy	8	9	10	best	0.94600
3680	entropy	12	9	10	best	0.94600
3684	entropy	12	9	14	best	0.94600
3686	entropy	12	9	16	best	0.94600
3688	entropy	12	9	18	best	0.94600
...	...	...	...	...	...	...
2771	entropy	2	7	18	random	0.87075
107	gini	2	11	18	random	0.87075
2773	entropy	2	9	2	random	0.87075
105	gini	2	11	16	random	0.87075
1	gini	2	1	2	random	0.87075

Best hyperparameters:
Criterion: Entropy
Max_depth: 8
Min_samples_leaf:9
Min_samples_split:10
Splitter: best

4.6. Decision Trees

Scaling Type	Correlated Variables	Cross-Validation	Accuracy	F1-score	Precision	Roc_auc
Standard	Yes	Yes	0.9480	0.7886	0.9238	0.8393
Standard	No	Yes	0.9490	0.7901	0.9412	0.8369
Minmax	Yes	Yes	0.9480	0.7886	0.9238	0.8393
Minmax	No	Yes	0.9490	0.7901	0.9412	0.8369

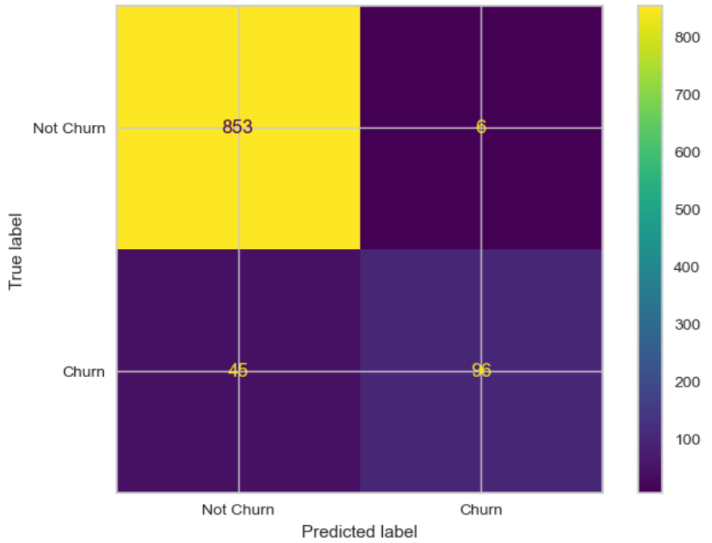
4.6. Decision Trees

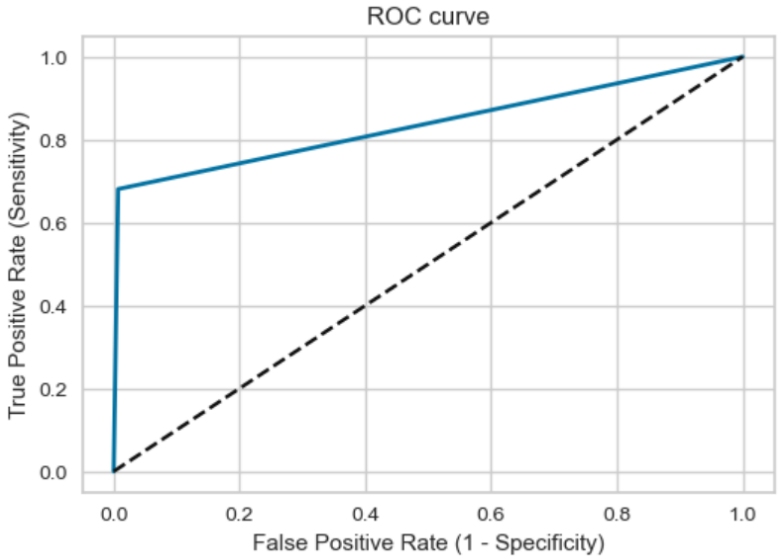


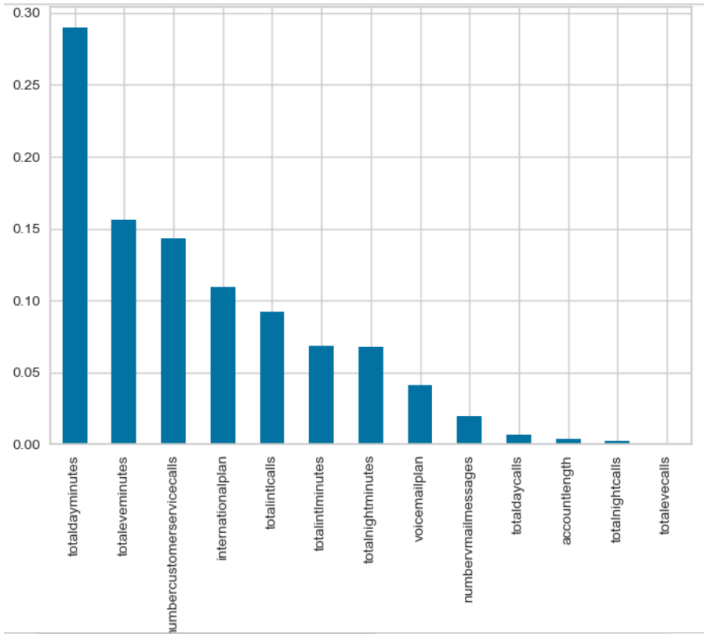
-Condition
-Entropy of the node
-Number of observations
-Number of observations present in each class

4.6. Decision Trees

Confusion matrix, roc curve and **feature_importance plot** for the test dataset using t he best model with
standard **scaling, without the correlated variables** and with **cross-validation**.

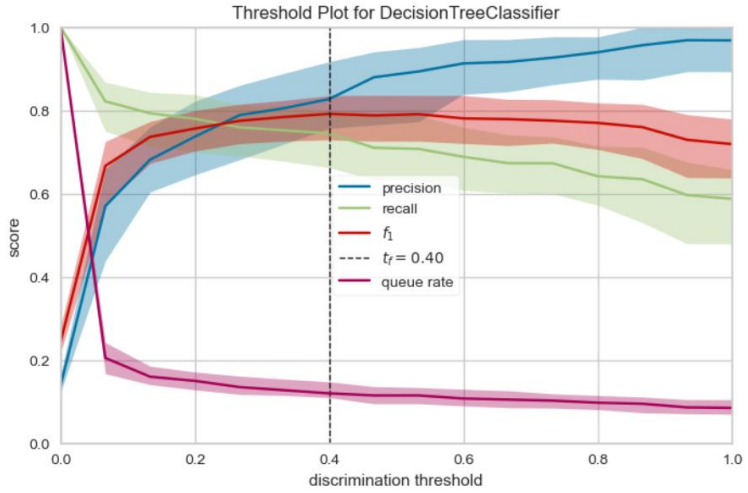


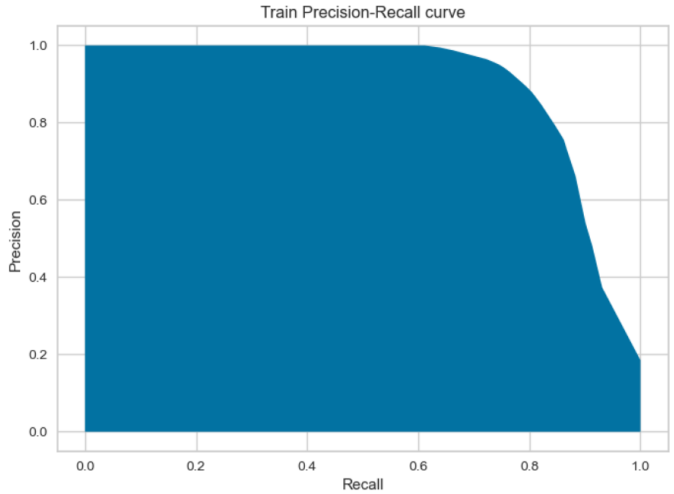


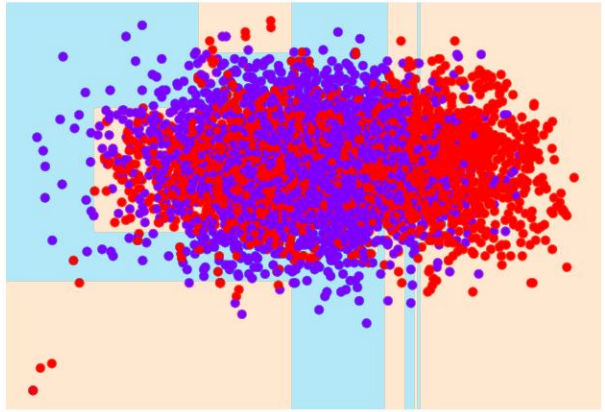


4.6. Decision Trees

Threshold plot, Precision-Recall curve and **a plot of two dimensions scatter plot** for the test dataset using the best model with **standard scaling, without the correlated variables** and with **cross-validation**.







4.7. Random Forest

	param_n_estimators	param_max_depth	mean_test_score
12	100	5	0.90450
15	100	5	0.90450
13	200	5	0.90100
16	200	5	0.90100
17	300	5	0.90075
14	300	5	0.90075
0	100	5	0.90075
3	100	5	0.90075
4	200	5	0.90000
1	200	5	0.90000
2	300	5	0.89950
5	300	5	0.89950
9	100	5	0.87425
6	100	5	0.87425
11	300	5	0.87350
8	300	5	0.87350

Scaling Type	Correlated Variables	Accuracy	F1-score	Precision	Roc_auc
Standard	Yes	0.9180	0.6466	0.8242	0.7566
Standard	No	0.9180	0.6466	0.8242	0.7566
Minmax	Yes	0.9180	0.6466	0.8242	0.7566
Minmax	No	0.9180	0.6466	0.8242	0.7566

Best hyperparameters:

Max_depth: 5

Max_features: 20

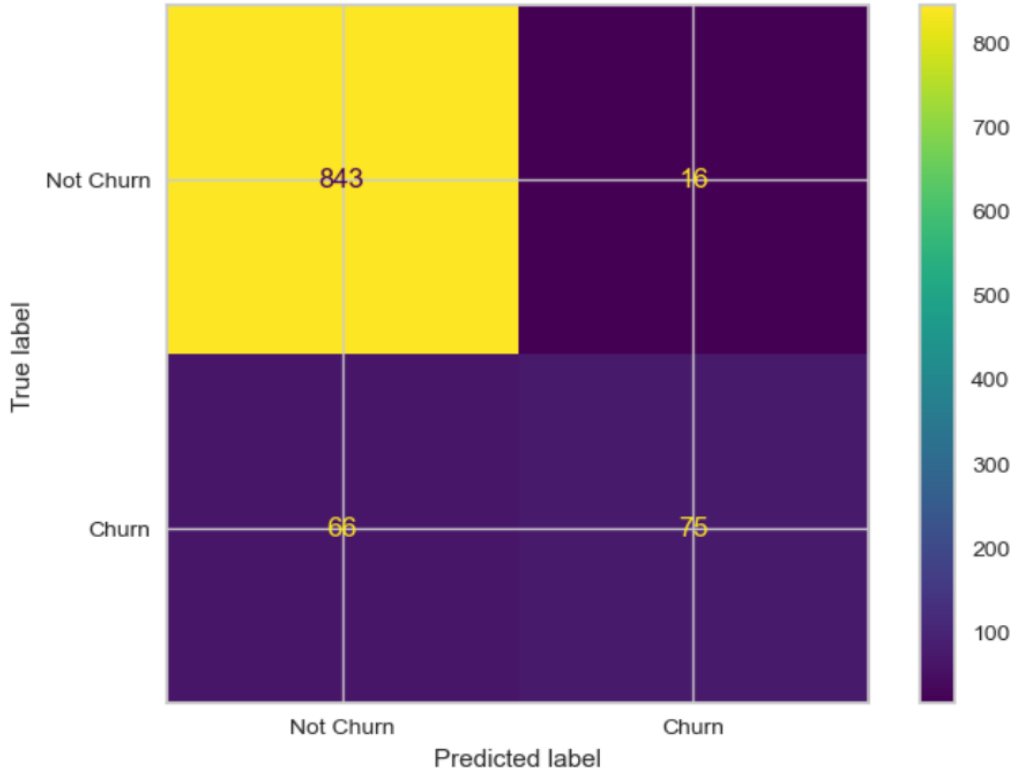
Min_samples_leaf: 50

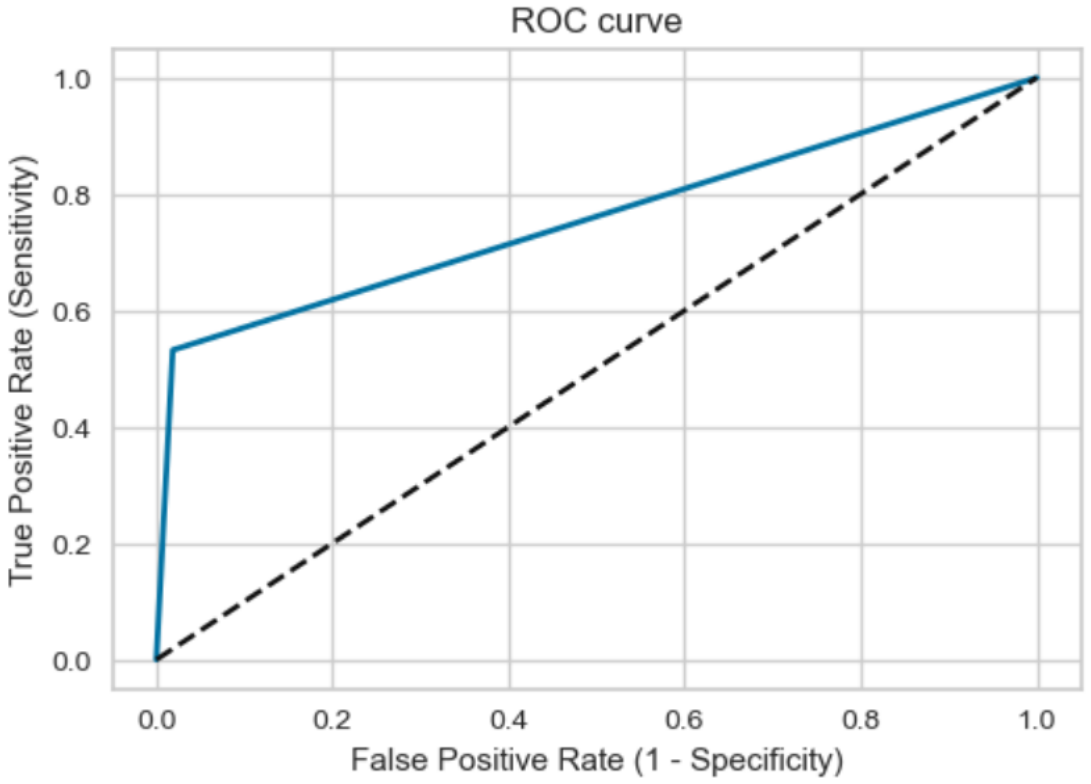
Min_samples_split: 50

N_estimators: 100

4.7. Random Forest

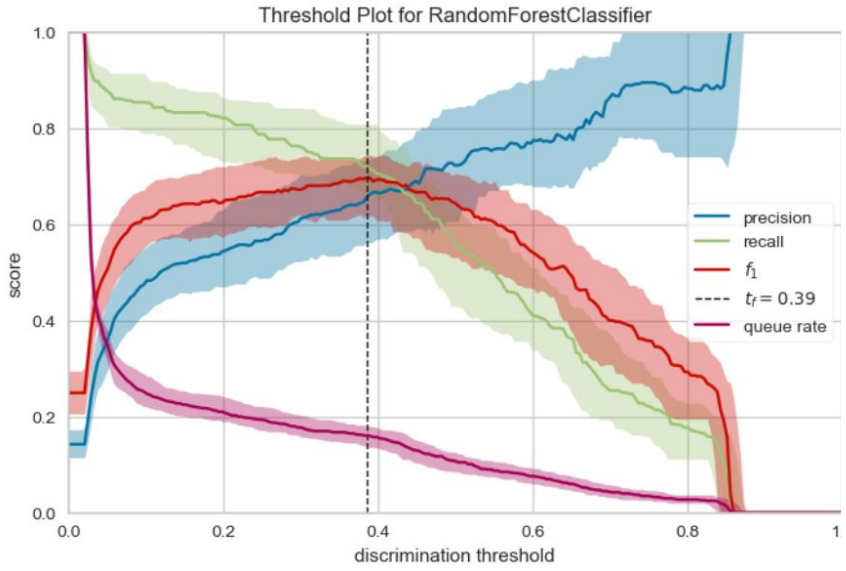
Confusion matrix
and roc curve for the test dataset using the best model with
standard scaling, with the correlated variables and with **cross-validation**.

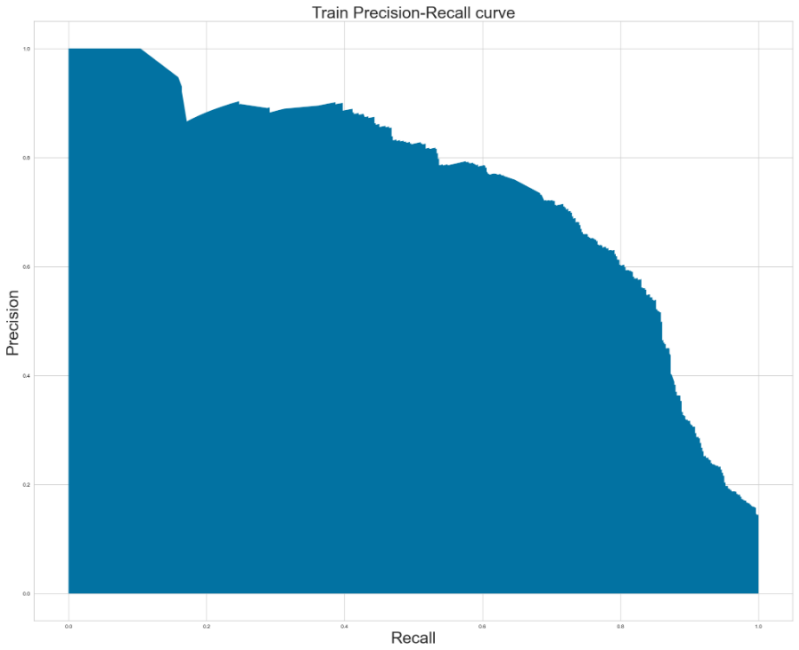


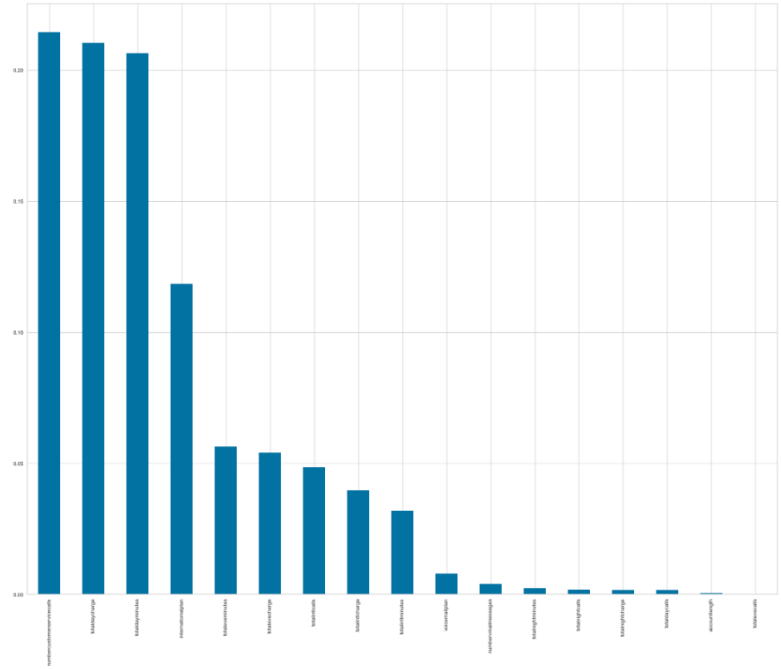


4.7. Random Forest

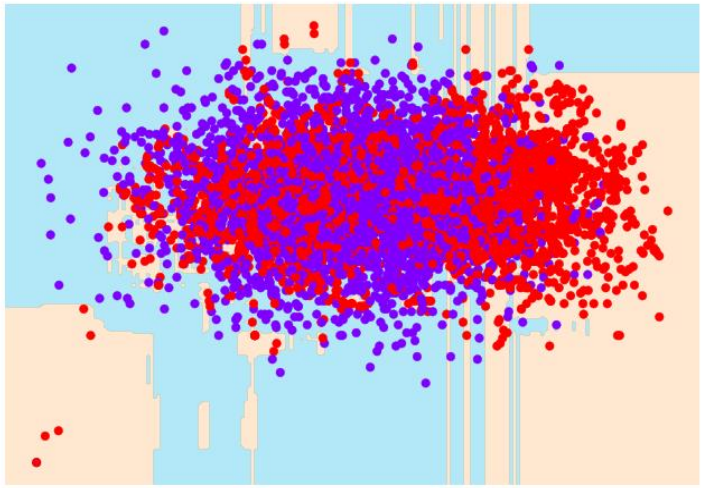
Threshold plot, Precision-Recall curve, feature_importance plot for the test dataset using the best model with **standard scaling, with the correlated variables** and with **cross-validation**.



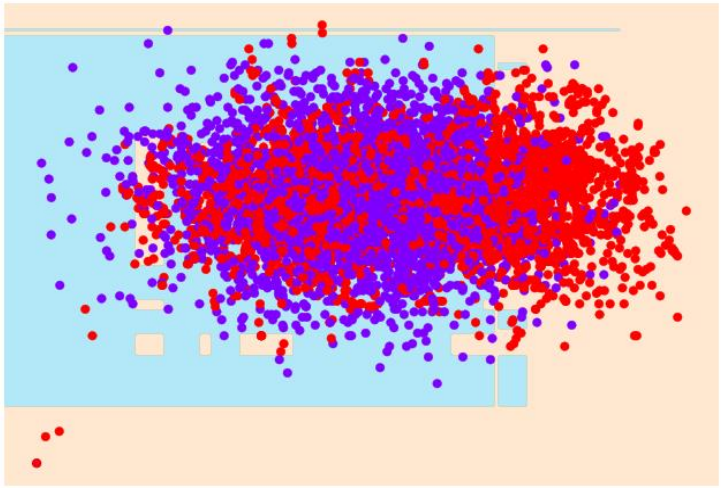




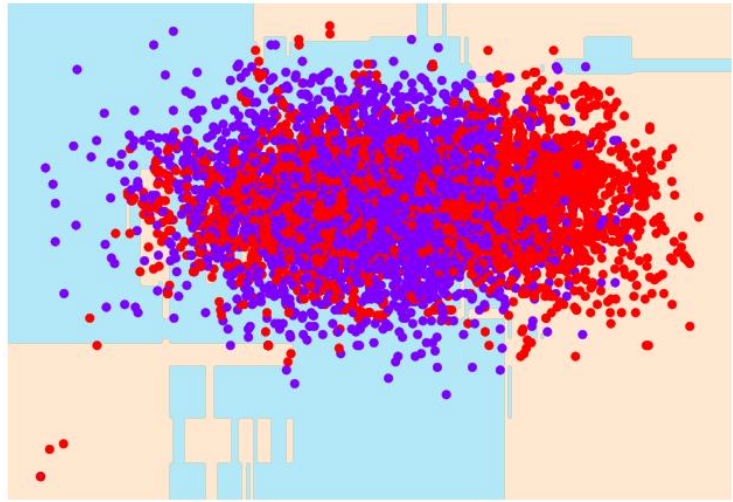
4.7. Random Forest



Random Forest



AdaBoost



Gradient Boosting

4.8. Support Vector Machines

	param_C	param_kernel	param_gamma	mean_test_score
19	5	rbf	0.1	0.92050
28	10	rbf	0.1	0.91950
10	1	rbf	0.1	0.91250
0	0.01	linear	0.1	0.85850
16	1	rbf	10	0.85850
34	10	rbf	10	0.85850
33	10	linear	10	0.85850
31	10	rbf	1	0.85850
30	10	linear	1	0.85850
27	10	linear	0.1	0.85850
25	5	rbf	10	0.85850
24	5	linear	10	0.85850
22	5	rbf	1	0.85850
21	5	linear	1	0.85850
1	0.01	rbf	0.1	0.85850

Best hyperparameters:

Kernel: rbf

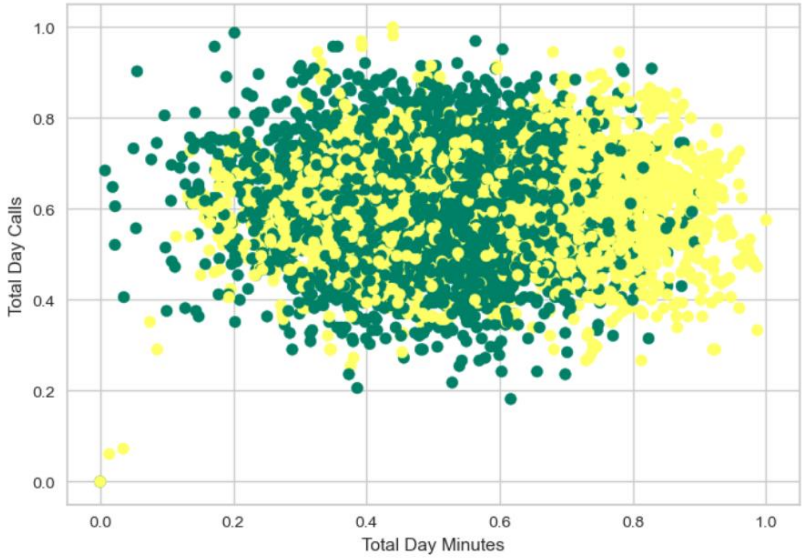
C=5

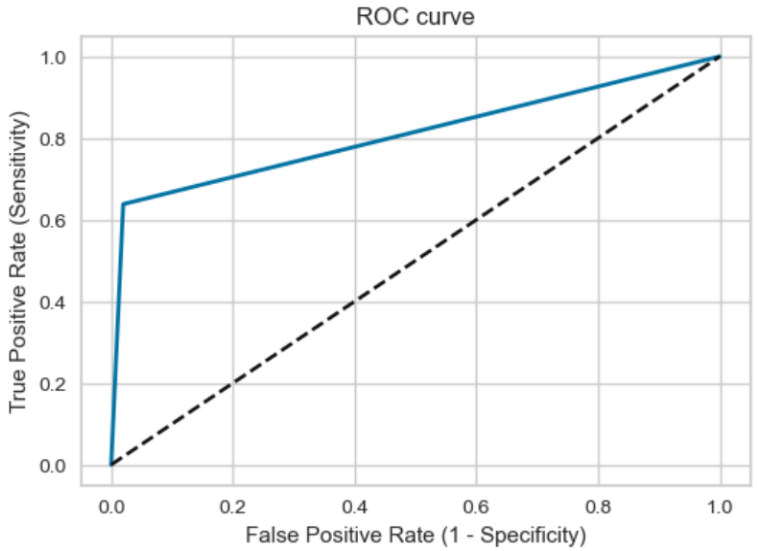
Gamma=0.1

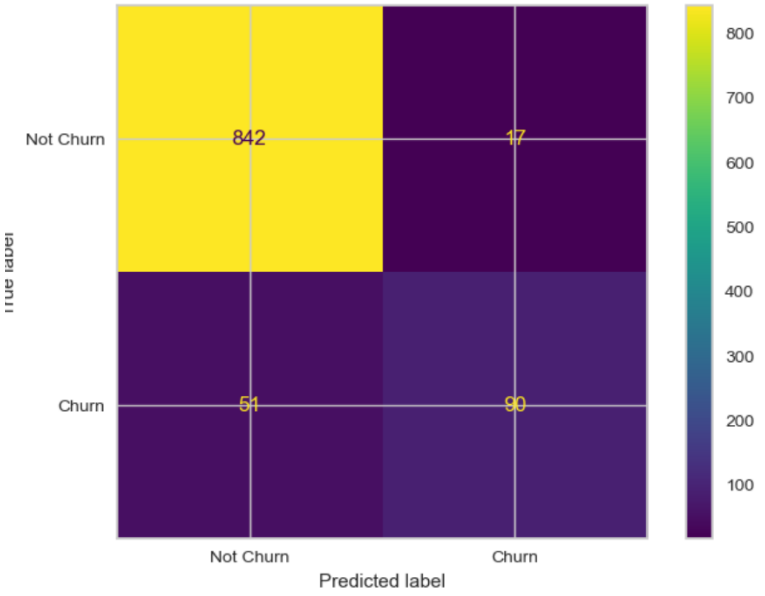
Scaling Type	Correlated Variables	Accuracy	F1-score	Precision	Roc_auc
Standard	Yes	0.9320	0.7258	0.8411	0.8093
Minmax	Yes	0.8730	0.1806	1.0	0.5496
Standard	No	0.9330	0.7265	0.8558	0.8069
Minmax	No	0.8610	0.0280	1.0	0.5071

4.8. Support Vector Machines

Plot of two dimensions scatter plot, r oc curve
and Confusion matrix for the test data set using the best model with **standard scaling, without the correlated variabl es** and with **cross-validation**.

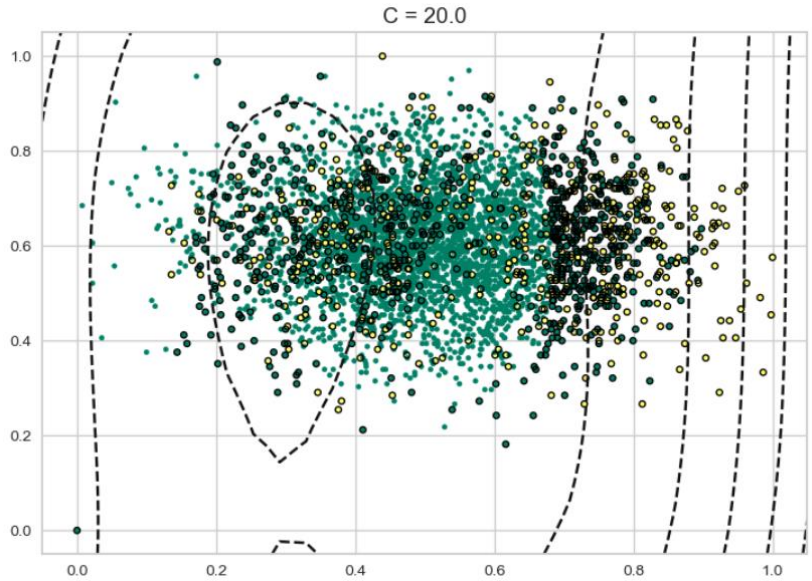


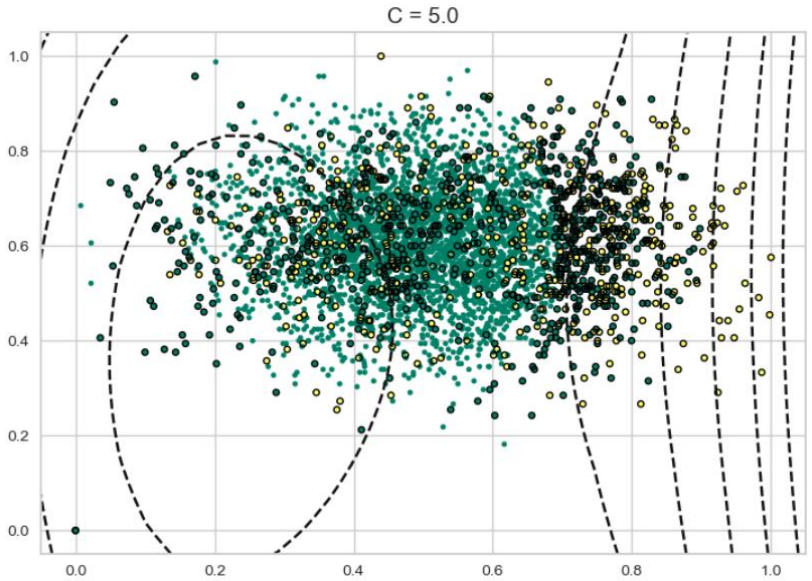


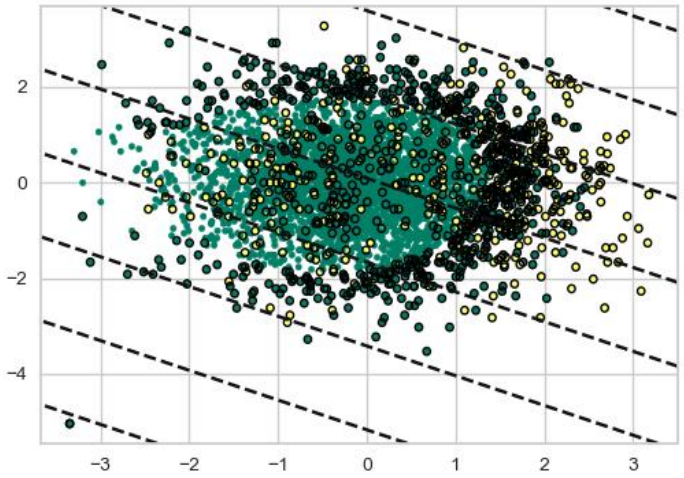


4.8. Support Vector Machines

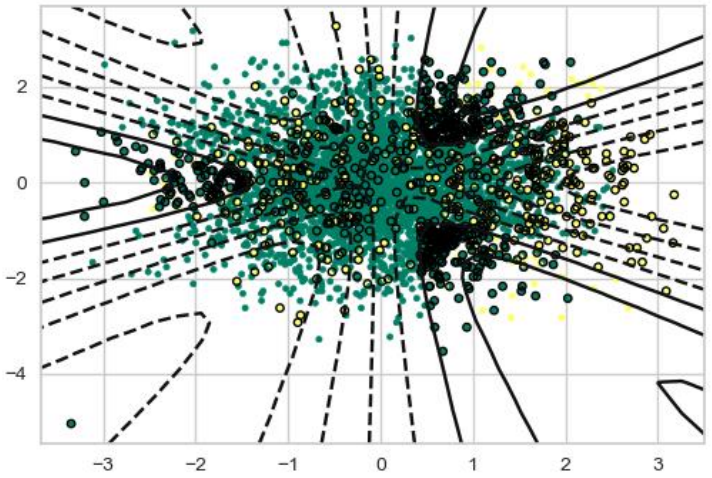
Kernel=Radial Basis Function(rbf)



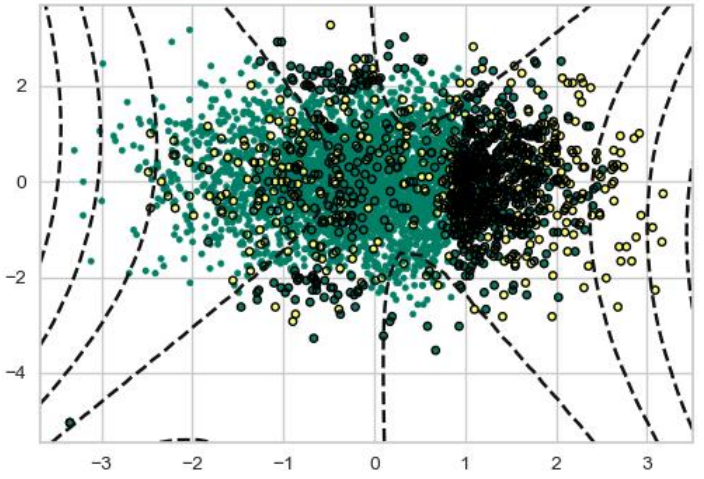




Kernel=Linear



Kernel=Sigmoid



Kernel=Polynomial (poly)

5. Handling imbalance datasets

Decision Tree

Resampling	Precision	Recall	F1-score	Accuracy
actual	0.922	0.667	0.774	0.945
smote	0.832	0.809	0.82	0.95
adasym	0.747	0.794	0.77	0.933
smote+tomek	0.767	0.794	0.78	0.937
smote+enn	0.673	0.816	0.737	0.918

Random Forest

Resampling	Precision	Recall	F1-score	Accuracy
actual	0.82	0.518	0.635	0.916
smote	0.714	0.851	0.777	0.931
adasym	0.649	0.801	0.717	0.911
smote+tomek	0.667	0.837	0.742	0.918
smote+enn	0.583	0.851	0.692	0.893

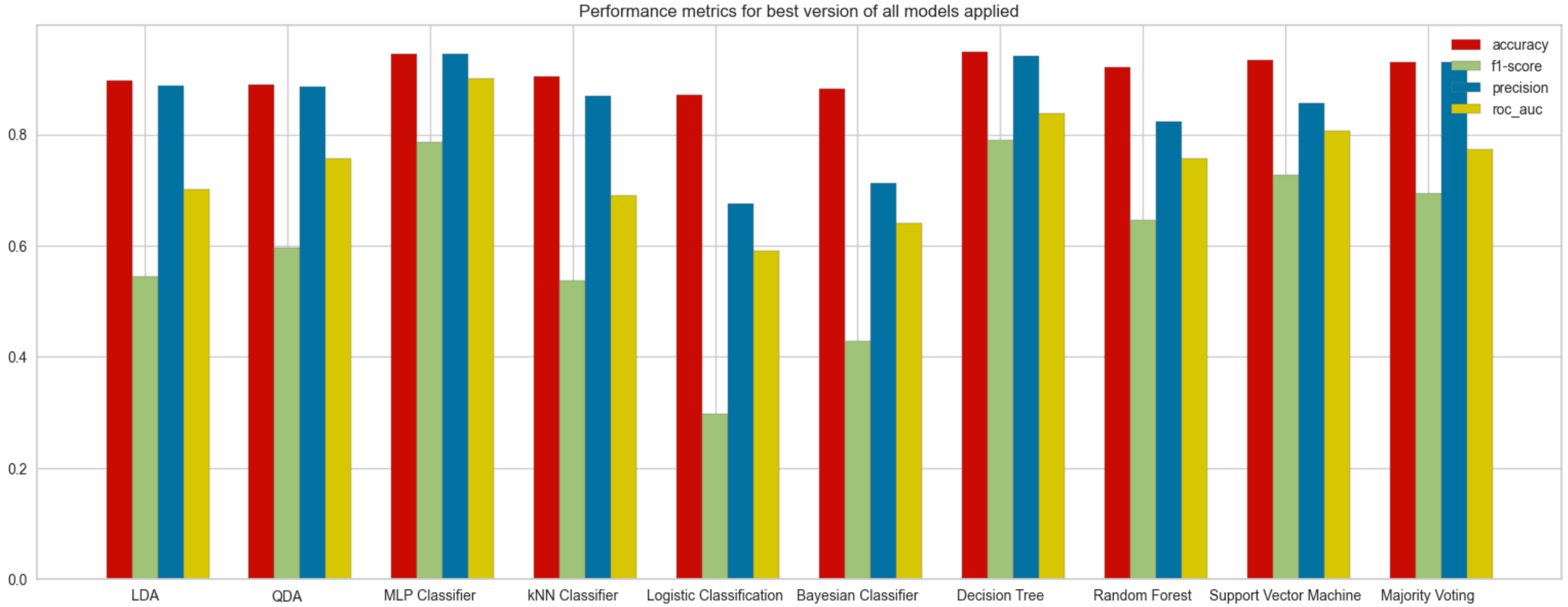
Support Vector Machine

Resampling	Precision	Recall	F1-score	Accuracy
actual	0.856	0.631	0.727	0.933
smote	0.639	0.667	0.653	0.9
adasym	0.608	0.681	0.642	0.893
smote+tomek	0.653	0.695	0.674	0.905
smote+enn	0.464	0.723	0.565	0.843

6. Evaluation and Main Conclusions

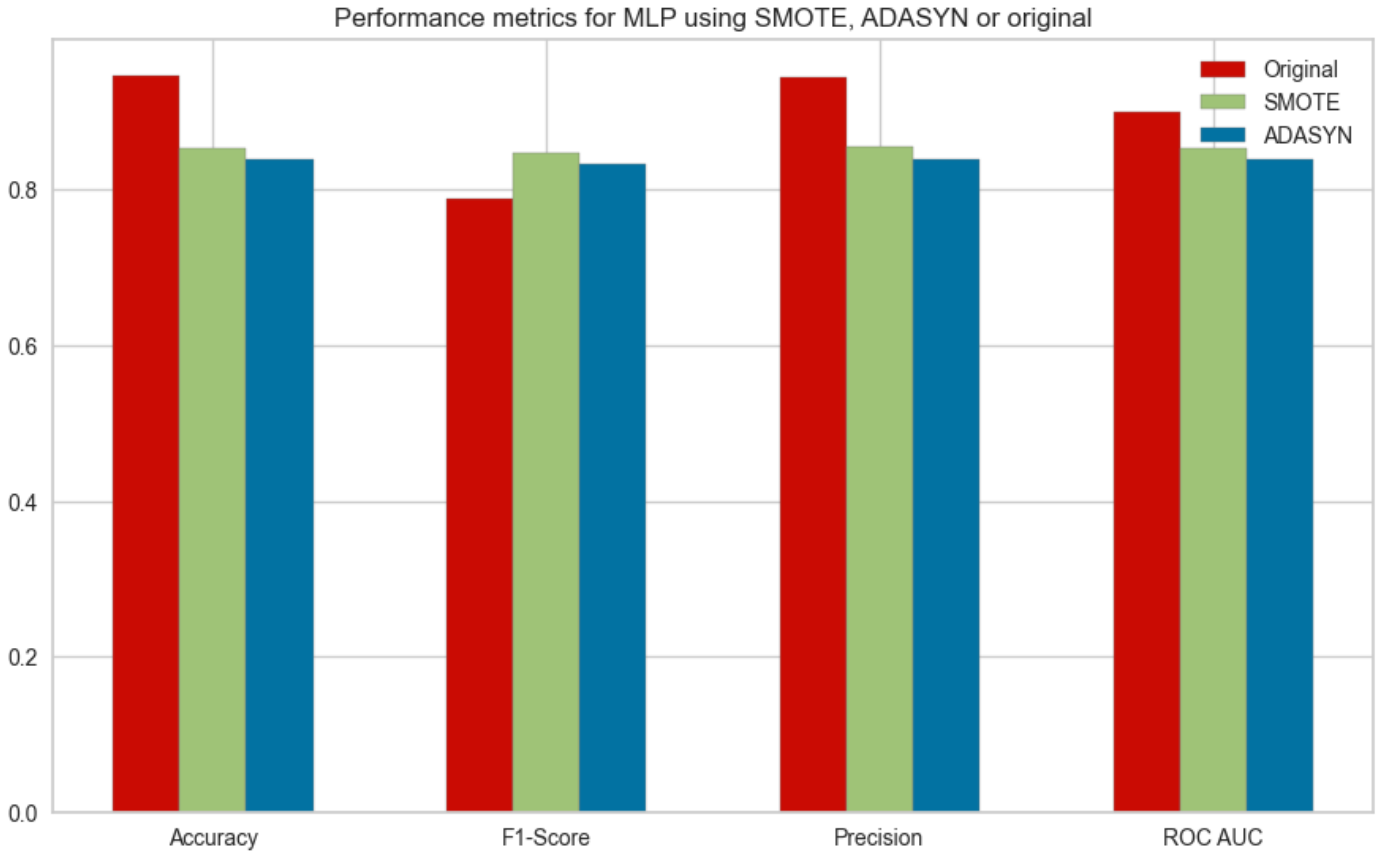
- **Model with overall best performance?** MLP Classifier

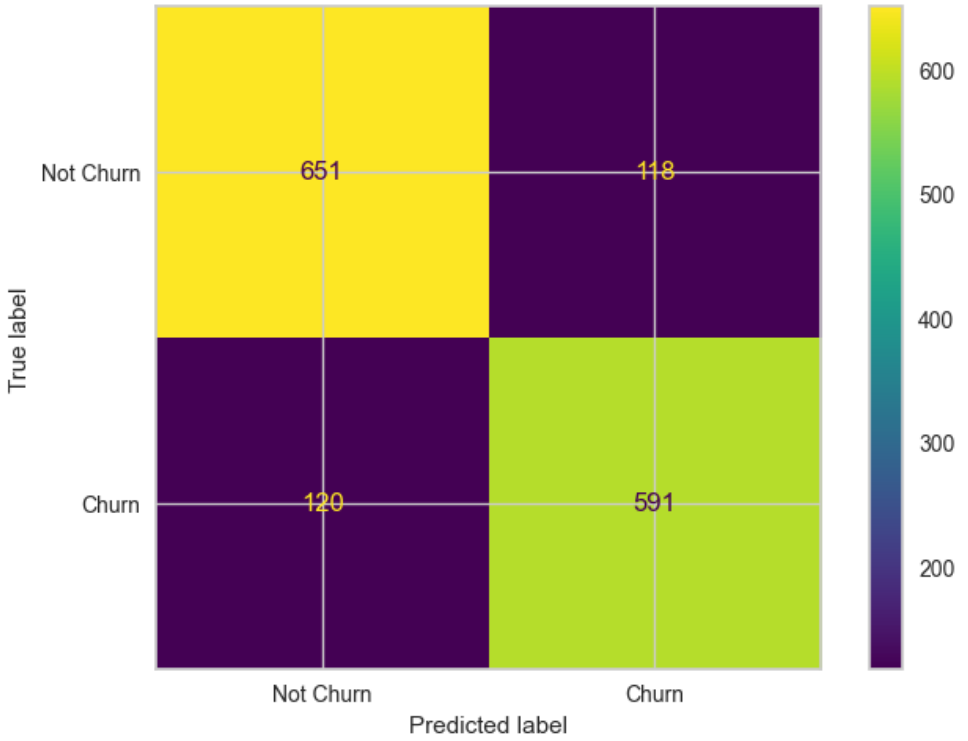
Alpha = 0.05
Solver = lbfgs
Hidden layer size = (9,)
Activation = relu



6. Evaluation and Main Conclusions

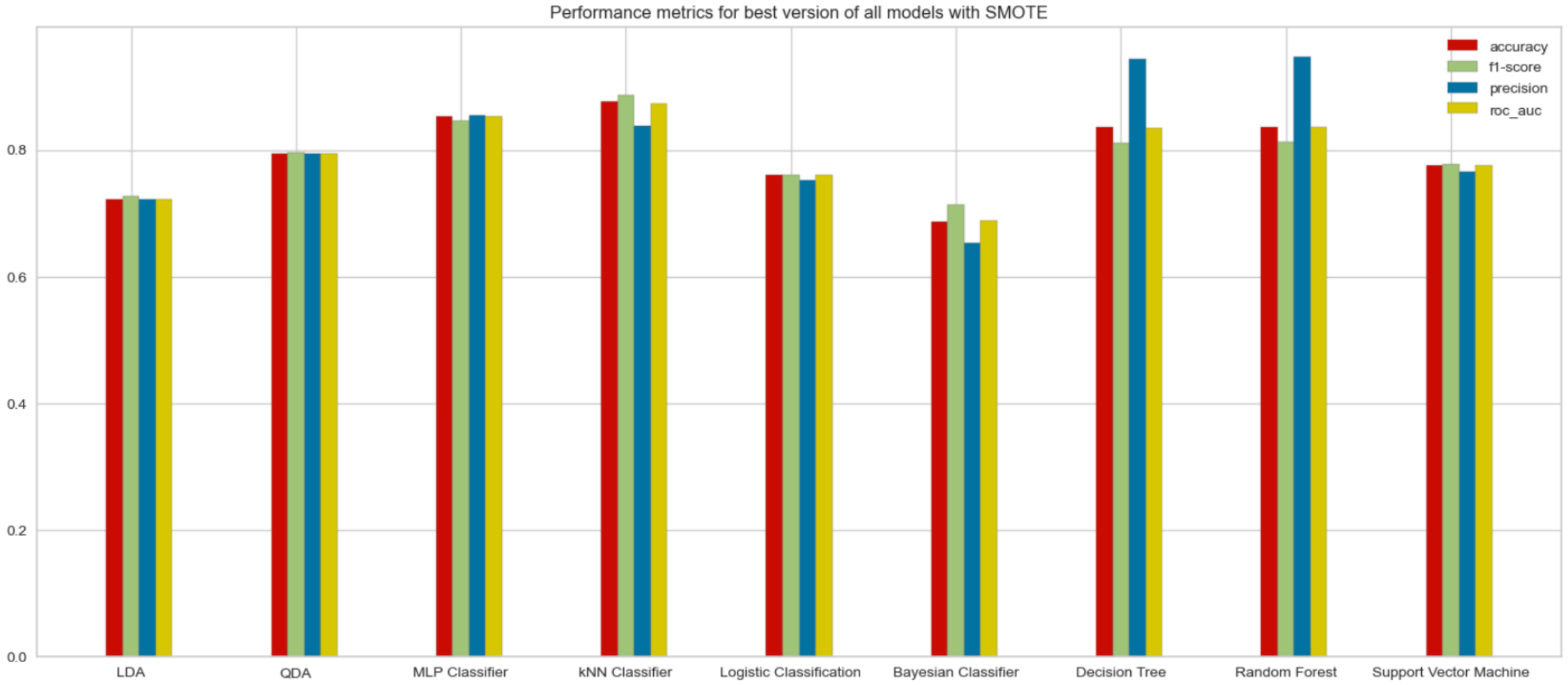
Applying re-sampling like SMOTE and ADASYN to the best model:





6. Evaluation and Main Conclusions

- **What are the main diferences observed with SMOTE?**



6. Evaluation and Main Conclusions

- **What are the lessons learnt?**
  - How to understand a business
  - How to understand the dataset that we want to study
  - How to prepare the data and use several models to classify it to achieve the business objective

- **What do you think that the business can gain from your data science effort?**
  - Our effort resulted in a model with good values of accuracy, F1-score, precision and ROC AUC, able to predict if a client is going to churn or not.
  - With it, the company will have a lot of information to confirm if some account is going to cancel (churn) and prevent it from happen.

- **If you had to explain to the colleagues of the marketing department how the model is doing its predictions which algorithm would you choose?**
  - k-Nearest Neighboors
- **And if you had to implement the most accurate model?**
  - Multi-layer perceptron.