Rapport de stage d’été volontaire

ANALYSE DE DONNEES ET APPRENTISSAGE AUTOMATIQUE AU SEIN DU DEPARATEMENT BI

1 JUILLET 2022

BANQUE NATIONALE AGRICOLE

Créé par : Ben Saber Rahma

Encadrant de stage : Mme Manaa Marwa

Introduction générale

Ce rapport est un compte rendu du stage d'été effectué au sein de la Banque Nationale Agricole, afin d'approfondir mes connaissances dans le domaine de l'analyse des données et de l'apprentissage automatique.

Problématique

Construire un modèle de Machine Learning capable de prédire la décision finale d'une demande de crédit déposée par un client de la BNA.

Choix méthodologique: la méthode CRISP-DM

1. Compréhension de la problématique
2. Compréhension des données
3. Préparation des données
4. Modélisation
5. Evaluation

Compréhension de la problématique

Ce projet s'insère dans le cadre de l'amélioration du processus de prise de décision lors du traitement d'une demande de crédit auprès de la banque. Nous viserons dans ce qui suit à automatiser cette procédure en utilisant un modèle d'apprentissage automatique capable de prédire la décision ACCORD ou bien REJET selon des facteurs différents.

Compréhension des données

La compréhension des données est une étape primordiale dans un projet Data Science. C'est lors de cette phase que l'on collecte, décrit, représente, explore et visualise les données, en vue de modéliser une solution optimale. Une compréhension approfondie des données permet d'éviter les problèmes que l'on pourra rencontrer lors des phases suivantes du projet.

Voici une vue d'ensemble de nos données, sous format xls:

Agence Initiatrice	N°CANEVAS	Type CANEVAS	Date Création	Code Produit	Montant Sollicité
017	F0171800002	CONSO	16/08/2018	1264	25 000 000
017	F0171800004	CONSO	16/08/2018	1264	30 000 000
017	F0171800005	CONSO	17/08/2018	1264	3 000 000
017	F0171800006	CONSO	17/08/2018	1264	10 000 000
017	F0171800007	CONSO	29/08/2018	1264	26 000 000
177	F1771800003	CONSO	29/08/2018	1259	8 000 000
083	F0831800007	CONSO	04/09/2018	1264	22 000 000
083	F0831800010	CONSO	04/09/2018	1264	9 000 000
100	F1001800001	CONSO	04/09/2018	1264	18 000 000
177	F1771800005	CONSO	04/09/2018	1264	26 000 000
177	F1771800006	CONSO	04/09/2018	1259	12 000 000
083	F0831800014	CONSO	12/09/2018	1264	20 000 000
083	F0831800016	CONSO	12/09/2018	1259	5 000 000
083	F0831800017	CONSO	12/09/2018	1264	12 000 000
146	F1461800004	CONSO	12/09/2018	1264	16 000 000

Figure 1 partie 1 du fichier Excel

Revenus Annuel	Retenus Mensuel	Profession
17 340 000	450 392	PROFESSION INCONNUE
21 699 924	0	PROFESSION INCONNUE
10 895 184	0	OUVRIERS QUALIFIES DE LA RECOLTE, DU CLASSEMENT ET DU STOCKAGE DE
14 818 788	0	PROFESSION INCONNUE
15 031 000	0	FORCES ARMEES
23 599 140	0	PROFESSION INCONNUE
23 424 000	0	PROFESSION INCONNUE
6 360 000	0	RETRAITES
10 949 112	0	AUTRES PERSONNEL DES SERVICES DIRECTS AUX PARTICULIERS, NON CLAS
15 444 000	0	AUTRES EMPLOYES DE BUREAU
24 516 924	0	AUTRES EMPLOYES DE BUREAU
13 176 000	0	SECRETAIRES
10 476 000	0	TECHNICIENS DES SCIENCES DE LA VIE
14 736 000	0	PROFESSIONS INTERMEDIAIRES DU TRAVAIL SOCIAL
19 914 000	0	PROFESSIONS INTERMEDIAIRES DU TRAVAIL SOCIAL
13 185 000	0	OFFICIERS MECANICIENS DE NAVIRES
6 672 000	0	RETRAITES
21 684 000	0	AGENTS D'ASSURANCES
6 216 000	0	RETRAITES
10 476 000	0	RETRAITES
4 956 000	156	RETRAITES
15 384 000	0	FORCES ARMEES

Figure 2 partie 2 du fichier Excel

Date Naissance	Sexe	Décision Finale	Centre Décision	Date MEP
13/01/1954	M	ACCORD	AGENCE	20/08/2018
12/04/1971	M	ACCORD	AGENCE	20/08/2018
24/04/1988	M	ACCORD	AGENCE	27/08/2018
10/09/1954	M	ACCORD	AGENCE	03/09/2018
22/09/1989	M	ACCORD	AGENCE	05/09/2018
20/08/1973	M	ACCORD	AGENCE	05/09/2018
18/06/1975	F	ACCORD	AGENCE	07/09/2018
04/10/1956	F	ACCORD	AGENCE	07/09/2018
11/04/1969	M	ACCORD	AGENCE	05/09/2018
20/12/1989	F	ACCORD	AGENCE	12/09/2018
09/10/1969	M	ACCORD	AGENCE	12/09/2018
10/01/1982	F	ACCORD	AGENCE	25/09/2018
30/07/1951	M	ACCORD	AGENCE	21/09/2018
28/07/1953	M	ACCORD	AGENCE	21/09/2018
16/04/1974	M	ACCORD	AGENCE	13/09/2018
22/04/1995	M	ACCORD	AGENCE	16/10/2018
02/03/1959	M	ACCORD	AGENCE	01/10/2018
03/07/1989	F	ACCORD	AGENCE	02/10/2018
01/07/1949	F	ACCORD	AGENCE	04/10/2018

Figure 3 partie 3 du fichier Excel

Données quantitatives continues

Les valeurs continues sont des valeurs qui s'expriment dans un intervalle de nombres réels infinis. Nous n'avons pas de valeurs quantitatives continues dans notre jeu de données.

Données quantitatives discrètes

Les valeurs discrètes sont des valeurs qui s'expriment dans un intervalle de nombres finis. Dans notre jeu de données, nous disposons des valeurs quantitatives discrètes suivantes: Agence initiatrice, code produit, montant sollicité, revenus mensuels, retenus mensuels.

Données qualitatives nominales

Les valeurs nominales sont des valeurs qualitatives exprimant le nom d'une catégorie : nom, sexe, métier, voiture, etc. Les données binaires sont des données catégorielles nominales, puisque généralement, une variable binaire représente deux valeurs conceptuellement opposées.

Dans notre jeu de données, nous disposons des valeurs qualitatives nominales suivantes: N°Canevas, Type canevas, Profession, Décision finale, Agence initiatrice, Sexe.

Données qualitatives ordinales

Les valeurs ordinales sont des valeurs qualitatives qui sont naturellement ordonnées et qui peuvent être traduites par une valeur numérique, comme le rang par exemple : élevé, moyen, bas. Nous n'avons pas de valeurs qualitatives ordinales dans notre jeu de données.

Données temporelles

Les données temporelles dans notre jeu de données sont des valeurs numériques qui suivent l'évolution du temps comme les suivantes: Date création, Date MEP, Date naissance.

Représentation des données

La représentation sert à modéliser les données en vue de s'assurer qu'elles sont adéquates à la méthode de travail ultérieure. Cela permet d'avoir une vision plus claire de la variable cible, de la variable à prédire, et de la façon par laquelle elles seront formatées dans les modèles d'apprentissage automatiques que nous allons utiliser.

Représentation des variables prédictives :

Les variables prédictives sont appelées variables indépendantes et sont largement connues dans la Data Science sous le nom anglais « Features». Dans une Data Set, les variables prédictives X= x1, x2, . . . , xn peuvent être modélisées dans une matrice d'ordre m*n où m est la taille du training set et n est le nombre des caractéristiques de chaque instance. Dans notre étude, n=15 et m=57302 comme l'explique la figure ci-dessous.

$$\begin{pmatrix} x(1,1) & \cdots & x(1,15) \\ \vdots & \ddots & \vdots \\ x(57302,1) & \cdots & x(57302,15) \end{pmatrix}$$

Représentation de la variable cible :

Bien entendu, la variable cible est la variable qu'on souhaite prédire à partir des variables prédictives. Elle est aussi appelée en anglais « Target variable ». Dans notre cas, nous sommes en train de prédire la décision de la banque suite à une demande de crédit, qui peut être soit ACCORD soit Rejet. Nous sommes dans en présence d'une variable cible unilabel binaire. Notre variable Cible est donc représentée par un vecteur de taille m=57302 qui est la taille du Dataset.

$$\begin{pmatrix} y(1) \\ . \\ . \\ . \\ y(57302) \end{pmatrix}$$

La préparation des données

Cette étape permet de faire ce qu'on appelle « Feature Engineering », pour avoir des attributs prêts pour le Machine Learning. Ensuite, elle permet d'identifier et de se débarrasser des valeurs erronées et dupliquées, de traiter les valeurs manquantes et de formater les types de données. Cela permet d'obtenir un « Training Set » plus fiable et structuré, de façon à ce que l'apprentissage automatique soit plus performant et soutienne une meilleure prise de décision.

Conversion du fichier de .xls à .csv :

Cette étape est primordiale puisque tout le travail effectué par la suite reposera sur la bibliothèque pandas de python, et qui fonctionne avec les fichiers csv.

Transformation des noms et types des attributs :

Cette étape est importante puisque certains algorithmes nécessitent que les attributs soient formatés d'une certaine manière. Il faut également s'assurer que les attributs ont des types corrects, et ce avec la fonction dtypes, puis procéder aux changements nécessaires.

```
Entrée [14]: df.dtypes

Out[14]: index
         Agence Initiatrice    object
         N°CANEVAS             object
         Type CANEVAS          object
         Date Création         object
         Code Produit          object
         Montant Sollicité     object
         Revenus Annuel        object
         Retenus Mensuel       object
         Profession            object
         Date Naissance        object
         Sexe                  object
         Décision Finale       object
         Centre Décision       object
         Date MEP              object
         dtype: object
```

Figure 4 types de données avant la transformation

```
Entrée [164]: df.dtypes

Out[164]: index
          Agence_Initiatrice             int32
          N°CANEVAS                     string
          Type_CANEVAS                  string
          Date_Création         datetime64[ns]
          Code_Produit                   int32
          Montant_Sollicité            float64
          Revenu_Annuel                float64
          Retenu_Mensuel               float64
          Date_Naissance        datetime64[ns]
          Sexe                          string
          Décision_Finale               string
          Centre_Décision               string
          Date_MEP              datetime64[ns]
          dtype: object
```

Figure 5 Types de données après la transformation

Ajout de la colonne Secteur

Dans notre dataset, nous avons 363 professions différentes. Nous avons donc décidé de traiter les professions par secteur : Libéral, employé, retraité et inconnu.

```
Entrée [ ]: df.loc[(df['Profession'] == 'PROFESSION INCONNUE') , 'Secteur'] = 'INCONNU'
df.loc[(df['Profession'] == 'RETRAITES') , 'Secteur'] = 'RETRAITES'
df.loc[(df['Profession'].str.contains('MEDECINS')), 'Secteur'] = 'LIBERAL'
df.loc[(df['Profession'].str.contains('MEDICALES')), 'Secteur'] = 'LIBERAL'
df.loc[(df['Profession'].str.contains('HUISSIER')), 'Secteur'] = 'LIBERAL'
df.loc[(df['Profession'].str.contains('NOTAIRES')), 'Secteur'] = 'LIBERAL'
df.loc[(df['Profession'].str.contains('PSY')), 'Secteur'] = 'LIBERAL'
df.loc[(df['Profession'].str.contains('ORTH')), 'Secteur'] = 'LIBERAL'
df.loc[(df['Profession'] == 'ARCHITECTES') & (df['Profession'] == 'SAGE-FEMMES')
       & (df['Profession'] == 'DIETETICIENS')  , 'Secteur'] = 'LIBERAL'
df.loc[(df['Secteur'] == '') , 'Secteur'] = 'EMPLOYE'
```

Figure 6 Création de l'attribut Secteur

Ajout de la colonne âge :

Nous avons ajouté la colonne âge à partir de la date de naissance des employés dans l'hypothèse que cela soit un facteur important dans la prise de décision.

Encodage des variables catégoriques :

One hot encoding avec la méthode get_dummies()

L'encodage one hot est une étape importante pour préparer votre ensemble de données en vue de son utilisation dans l'apprentissage automatique. Cette technique transforme les données catégorielles en une représentation vectorielle binaire.

Cela signifie que pour chaque valeur unique dans une colonne, une nouvelle colonne est créée. Les valeurs de cette colonne sont représentées par des 1 et des 0, selon que la valeur correspond ou non à l'en-tête de colonne.

Nous avons utilisé cette technique pour les variables Secteur, Type_canevas et Sexe.

typecanevas_AUTO	typecanevas_CONSO	typecanevas_IMMO
0	1	0
0	1	0
0	1	0
0	1	0
0	1	0

Figure 7 exemple de one hot encoding sur la variable typecanevas

Label encoding :

Cette technique permet d'encoder les étiquettes cibles avec une valeur comprise entre 0 et n_classes-1. Ce transformateur doit être utilisé pour coder les valeurs cibles, c'est-à-dire y, et non l'entrée X.

Nous avons donc utilisé le label encoder pour notre variable prédictive Décision_finale.

Décision_Finale
0
0
0
0
0

Figure 8 Label encoder sur la variable Décision_Finale

Le nettoyage des valeurs nulles

Les valeurs nulles donnent des résultats erronés dans l'apprentissage automatique et certains algorithmes ne travaillent que si toutes les valeurs nulles du training set sont supprimées. Cette étape est réalisée avec la méthode dropna().

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 57298 entries, 2 to 57299
Data columns (total 14 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   Agence_Initiatrice  57298 non-null  int32
 1   N°CANEVAS           57298 non-null  string
 2   Type_CANEVAS        57298 non-null  string
 3   Date_Création       57298 non-null  datetime64[ns]
 4   Code_Produit        57298 non-null  int32
 5   Montant_Sollicité   57298 non-null  float64
 6   Revenu_Annuel       57298 non-null  float64
 7   Retenu_Mensuel      57298 non-null  float64
 8   Profession          57298 non-null  string
 9   Date_Naissance      57290 non-null  datetime64[ns]
 10  Sexe                57288 non-null  string
 11  Décision_Finale     57298 non-null  string
 12  Centre_Décision     57298 non-null  string
 13  Date_MEP            45986 non-null  datetime64[ns]
dtypes: datetime64[ns](3), float64(3), int32(2), string(6)
memory usage: 6.1 MB
```

Figure 9 résultat de dropna

Le nettoyage des lignes dupliquées

Le nettoyage des valeurs dupliquées se fait avec la fonction drop_dup() de Pandas.

```
: df.duplicated().sum()
: 0
```

Figure 10 Nombre de lignes dupliquées

Fichier csv résultant :

Notre Data Set est maintenant prête à être proprement exploitée.

jupyter retailsfinal.csv 24/07/2022 Logou

le Edit View Language current mo

```
N°CANEVAS,Date_Création,Code_Produit,Montant_Sollicité,Revenu_Annuel,Retenu_Mensuel,Date_Naissance,Décision_Finale,Centre_Décision,secteu
r_EMPLOYE,secteur_INCONNU,secteur_LIBERAL,secteur_RETRAITES,typecanevas_AUTO,typecanevas_CONSO,typecanevas_IMMO,sexe_F,sexe_M,age
0,0,0,2,F0171800002,2018-08-16,1264,25000000,17340000,450392,1954-01-13,0,AGENCE,0,1,0,0,0,1,0,0,1,64
1,1,1,3,F0171800004,2018-08-16,1264,30000000,21699924,0,1971-12-04,0,AGENCE,0,1,0,0,0,1,0,0,1,46
2,2,2,4,F0171800005,2018-08-17,1264,3000000,10895184,0,1988-04-24,0,AGENCE,1,0,0,0,0,1,0,0,1,30
3,3,3,5,F0171800006,2018-08-17,1264,10000000,14818788,0,1954-10-09,0,AGENCE,0,1,0,0,0,1,0,0,1,63
4,4,4,6,F0171800007,2018-08-29,1264,26000000,15031000,0,1989-09-22,0,AGENCE,1,0,0,0,0,1,0,0,1,28
5,5,5,7,F1771800003,2018-08-29,1259,8000000,23599140,0,1973-08-20,0,AGENCE,0,1,0,0,0,1,0,0,1,45
6,6,6,8,F0831800007,2018-04-09,1264,22000000,23424000,0,1975-06-18,0,AGENCE,0,1,0,0,0,1,0,1,0,42
7,7,7,9,F0831800010,2018-04-09,1264,9000000,6360000,0,1956-04-10,0,AGENCE,0,0,0,1,0,1,0,1,0,61
8,8,8,10,F1001800001,2018-04-09,1264,18000000,10949112,0,1969-11-04,0,AGENCE,1,0,0,0,0,1,0,0,1,48
9,9,9,11,F1771800005,2018-04-09,1264,26000000,15444000,0,1989-12-20,0,AGENCE,1,0,0,0,0,1,0,1,0,28
10,10,10,12,F1771800006,2018-04-09,1259,12000000,24516924,0,1969-09-10,0,AGENCE,1,0,0,0,0,1,0,0,1,48
11,11,11,13,F0831800014,2018-12-09,1264,20000000,13176000,0,1982-10-01,0,AGENCE,1,0,0,0,0,1,0,1,0,36
12,12,12,14,F0831800016,2018-12-09,1259,5000000,10476000,0,1951-07-30,0,AGENCE,1,0,0,0,0,1,0,0,1,67
13,13,13,15,F0831800017,2018-12-09,1264,12000000,14736000,0,1953-07-28,0,AGENCE,1,0,0,0,0,1,0,0,1,65
14,14,14,16,F1461800004,2018-12-09,1264,16000000,19914000,0,1974-04-16,0,AGENCE,1,0,0,0,0,1,0,0,1,44
15,15,15,17,F0171800010,2018-09-17,1264,23000000,13185000,0,1995-04-22,0,AGENCE,1,0,0,0,0,1,0,0,1,23
16,16,16,18,F0831800018,2018-09-19,1264,5000000,6672000,0,1959-02-03,0,AGENCE,0,0,0,1,0,1,0,0,1,59
17,17,17,19,F0831800023,2018-09-27,1264,12000000,21684000,0,1989-03-07,0,AGENCE,1,0,0,0,0,1,0,1,0,29
18,18,18,20,F0831800024,2018-09-27,1259,2000000,6216000,0,1949-01-07,0,AGENCE,0,0,0,1,0,1,0,1,0,69
```

Figure 11 Dataset finale

Modélisation

La modélisation est l'étape de la méthodologie CRISP-DM que la plupart des Data scientistes préfèrent le plus. Les données sont maintenant bien structurées et prêtes à être modélisées en vue de résoudre la problématique. Dans cette phase, le praticien doit sélectionner les techniques de modélisation et les algorithmes à essayer, générer une conception de test pour diviser la Data Set en une partie pour l'apprentissage et une partie pour le test, et enfin construire le modèle.

Choix de la technique de prédiction adéquate

Afin de choisir la technique de prédiction adéquate à son problème, il faut se poser les questions suivantes :

• Les performances de calcul sont-elles un problème ?

Si oui, il est préférable de :

— Réduire la dimensionnalité.

— Utiliser des algorithmes peu couteux.

— Sélectionner seulement les attributs nécessaires à la prédiction.

— Choisir des algorithmes appelés « Lazy Learners » comme KNN.

• Quel est le type de ma variable cible ?

Le type de la variable cible est presque définitif dans le choix de la technique de prévision pour Un problème d'apprentissage automatique. En effet, il est largement reconnu que :

— Quand la variable à prédire est continue : Il s'agit d'un problème de régression.

— Quand la variable à prédire est catégoriale (nominale) : Il s'agit d'un problème de classification, qui est notre cas.

— Quand la variable à prédire est ordinale : Il s'agit d'un problème de classification classée.
— Pas de variable à prédire, le but est de trouver une structure dans les données : Il s'agit d'un problème de clustering, Projection.
• Est-ce que les données sont linéairement séparables ?
La réponse à cette question est dure à connaitre en amont. Pour remédier à cette contrainte d'incertitude, il est préférable de tester plusieurs modèles d'apprentissage automatique et de faire une étude comparative pour en ressortir celui qui s'ajuste le mieux à la Data Set.

Dataset déséquilibrée et Tuning des Hyper paramètres

Lors de la visualisation des données, nous avons compris que notre Data Set est « Imbalanced » qui est le mot anglais pour déséquilibrée. Dans notre cas, nous avons beaucoup plus d'échantillons avec l'output 0 qu'avec l'output 1. Pour résoudre ce problème, nous avons implémenté la méthode de recherche de meilleurs paramètres suivante : La Grid Search
Cette fonction va estimer le poids de chaque point de donnée, en considérant le déséquilibre mentionné. D'autant plus, dans la phase d'évaluation, nous allons nous concentrer sur une mesure de performance différente que celle des problèmes de classification non déséquilibrés.

Algorithmes de Machine Learning utilisés :

Logistic Regression

En apprentissage automatique, la régression logistique est un type de modèle de classification paramétrique. Cela veut dire que ce modèle a un nombre fixe de paramètres qui dépend du nombre des attributs passés en input. Le résultat de ce modèle est une prédiction catégoriale.

```
lg3 = LogisticRegression(random_state=13)
# define evaluation procedure
grid = GridSearchCV(lg3,hyperparam_grid,scoring="roc_auc", cv=100, n_jobs=-1, refit=True)
grid.fit(x_train,y_train)
```

Figure 12 classifieur de régression logistique

K Nearest Neighbors

L'algorithme KNN classifie les points de données en se basant sur ceux qui leur sont le plus similaires. Dans la figure qui suit, le point x est comparé aux points les plus proches et les plus similaires. La distance entre le point x et le point le plus proche du groupe rouge, du groupe vert et du groupe bleu est calculée.

```
knn = KNeighborsClassifier()
k_range = list(range(1, 31))
param_grid = dict(n_neighbors=k_range)
grid = GridSearchCV(knn, param_grid, cv=10, scoring='roc_auc', return_train_score=False,verbose=1)
grid_search=grid.fit(x_train, y_train)
```

Figure 13 classifieur KNN

Naïve Bayes

Naïve Bayes est un groupe d'algorithmes d'apprentissage automatique supervisé basé sur le théorème de Bayes. C'est une technique simple de classification, mais qui est très fonctionnelle et performante. Ces algorithmes sont particulièrement utilisés quand la dimensionnalité des données est importante. Même les problèmes de classification complexes peuvent être implémentés et résolus grâce au classificateur Naïve Bayes.

```
nbModel_grid = GridSearchCV(estimator=GaussianNB(), scoring="roc_auc", param_grid=param_grid_nb, verbose=1, cv=10, n_jobs=-1)
nbModel_grid.fit(X_train, y_train)
```

Figure 14 classifieur Naïve Bayes

Random Forest

Les forets aléatoires est un algorithme d'apprentissage automatique supervisé, utilisé dans des problèmes de classification et de régression. Cet algorithme construit un ensemble d'arbres de décision et les fusionne ensemble pour obtenir une prédiction plus stable et précise. Il fonctionne avec la méthode « bagging » : une approche ensembliste qui améliore le résultat obtenu.

```
rfr = RandomForestRegressor(random_state = 1)
g_search = GridSearchCV(estimator = rfr, param_grid = param_grid,

                        cv = 3, n_jobs = 1, verbose = 0, return_train_score=True)
```

Figure 15 classifieur Random Forest

Evaluation

Dans les phases précédentes du CRISP-DM, nous avons exploré et préparé nos données, puis nous les avons modélisées à travers différents algorithmes et suivant la méthode Grid Search. Dans cette phase, nous allons répondre aux questions suivantes: Les modèles que nous avons construits, sont-ils performants ? répondent-ils à l'objectif final de notre projet ? Pour cela, nous devons faire une évaluation détaillée de notre travail à partir d'une étude comparative de tous les algorithmes implémentés, pour en ressortir le meilleur en termes de performance et d'efficacité.

La mesure de performance ROC :

La plupart des problèmes de classification déséquilibrés impliquent deux classes : un cas négatif avec la majorité des exemples et un cas positif avec une minorité d'exemples. Deux outils de diagnostic qui aident à l'interprétation des modèles prédictifs de classification binaire (à deux classes) sont les courbes ROC et les courbes de précision-rappel.
Nous avons choisi d'évaluer nos modèles avec la mesure de performance ROC.

Algorithme	KNN	Logistic Regression	Random Forest	Naive Bayes
ROC	0.667	0.672	0.13	0.638
Accuracy	0.93	0.93	0.93	0.93

Figure 16 Evaluation des modèles

Le meilleur algorithme en terme de performance est le Logistic Regression.

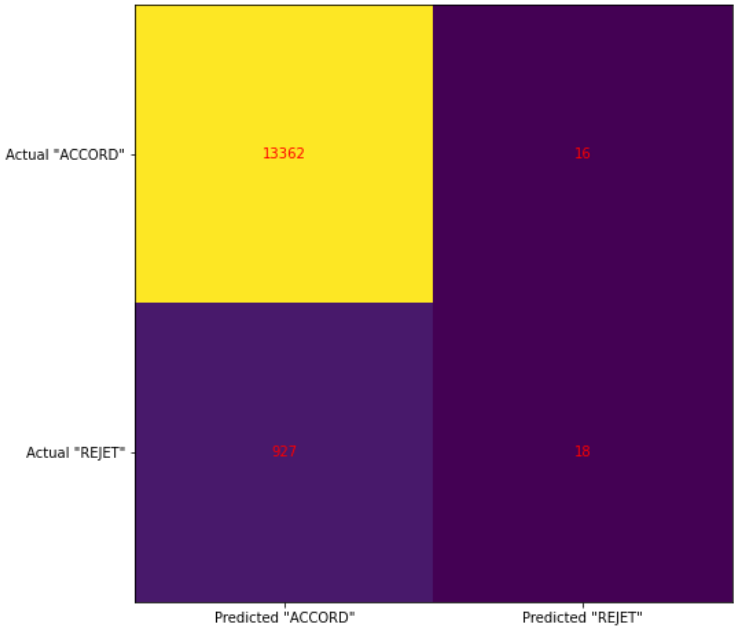

Figure 17 Matrice de confusion pour l'algorithme Logistic Regression

Conclusion générale :

Ainsi, dans le cadre de mes études en Big Data, j'ai eu la chance d'effectuer un stage d'été volontaire au sein de la banque BNA, dans son département BI. Cette expérience a été très enrichissante car elle m'a permis de découvrir en contexte le secteur bancaire, ainsi que les différentes missions et postes qui contribuent à son développement, surtout dans l'activité informatique. Aujourd'hui, après un mois dans l'entreprise, j'ai pu acquérir de nouvelles compétences et soulever de nouveaux défis posés par ce projet. Néanmoins, ce travail présente plusieurs persperctives d'amélioration, dont la plus importante est l'amélioration de résultats de classification, qui revient à consulter un expert du domaine métier et à essayer de collectionner des données encore plus pertinentes.