

Construction d'un score de crédit à partir du modèle de Probabilité de Défaut (PD).

Après avoir calculé notre modèle PD précédemment, nous continuons avec la construction d'une notation de crédit qui nous permet de simplifier l'interprétation des résultats. Rappelons qu'en vertu des exigences réglementaires contenues dans les accords de Bâle, ces résultats doivent être facilement compréhensibles pour les clients de la banque. Nous appliquons ici la méthodologie proposée par le Dr. Carlos Martinez dans le cadre de la formation sur l'analyse des risques de crédit.

Tout d'abord, en utilisant la matrice de confusion, la courbe ROC, l'AUC et les coefficients de Gini et Kolmogorov-Smirnov, nous reverrons la performance obtenue par le modèle PD calculé.

Matrice de confusion

```
In [121]: VP = (pd.crosstab(df_predict['Real'], df_predict['Prediction'],  
rownames = ['Reel'], colnames = ['PredicT']) / df_predict.shape[0]).iloc[1,1]  
VP
```

```
Out[121]: 0.9060027021810462
```

```
In [122]: VN = (pd.crosstab(df_predict['Real'], df_predict['Prediction'],  
rownames = ['Reel'], colnames = ['PredicT']) / df_predict.shape[0]).iloc[0,0]  
VN
```

```
Out[122]: 2.1445881318492783e-05
```

```
In [123]: Accuracy = VP + VN
```

```
In [124]: Accuracy
```

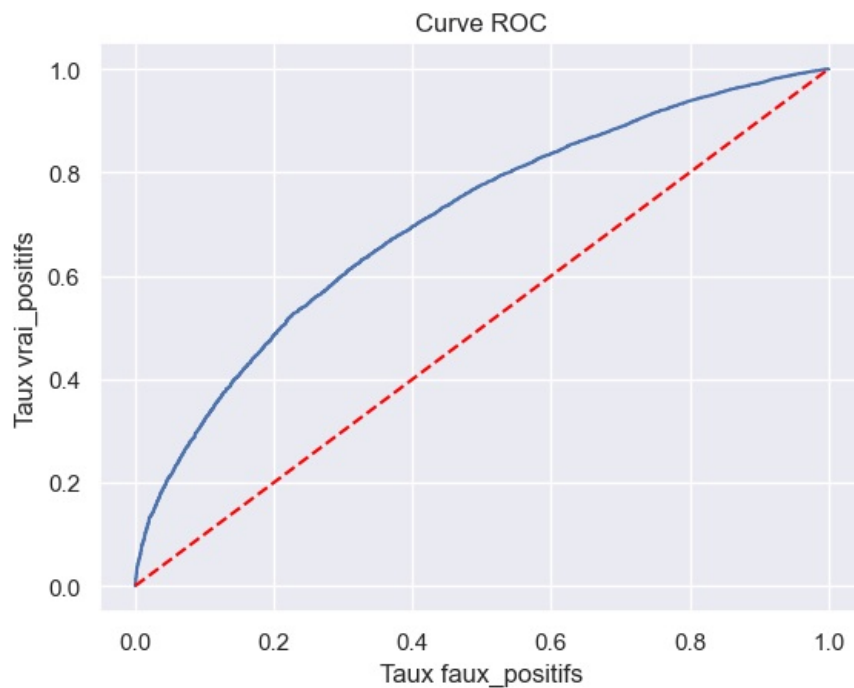
```
Out[124]: 0.9060241480623646
```

Le modèle a une précision relativement élevée de 90 % et fait un excellent travail pour prédire les clients bons. Cependant, il a une mauvaise performance pour prédire les clients mauvais. Cela s'explique par le fait que la majorité des observations dans la base sont des clients bons. Par conséquent, le modèle avec un seuil de coupure de 0,5 aura tendance à prédire que la plupart sont des clients bons, générant ainsi un grand nombre de faux positifs. Imaginons maintenant que nous utilisons ce modèle et ce seuil de coupure pour accorder des crédits : nous accorderions ainsi du crédit à de nombreux clients mauvais. Pour cette raison, il est nécessaire d'établir un seuil de coupure plus conservateur, disons 0,85. Recalculons la matrice de confusion avec un seuil. Avec ce nouveau seuil de coupure, nous réduirions considérablement le nombre de défauts, mais aussi le nombre de demandes approuvées. Alors, il faut trouver un seuil optimal qui minimise le coût financier.

ROC y AUC

```
In [131]: plt.plot(faux_positifs, vrai_positifs) # Nous définissons les données du graphique, avec des valeurs de x fictives  
plt.plot(faux_positifs, faux_positifs, linestyle = '--', color = 'red') # Benchmark (Predictor 50/50) la diagonale  
plt.xlabel('Taux faux positifs')  
plt.ylabel('Taux vrai positifs')  
plt.title('Curve ROC')
```

```
Out[131]: Text(0.5, 1.0, 'Curve ROC')
```



In [132... *#La métrique AUC (aire sous la courbe) nous sert à évaluer la qualité du modèle. Plus écartées sont les deux li*

In [133... `AUC = roc_auc_score(df_predict['Real'], df_predict['Proba_Prediction'])`

In [134... `AUC`

Out[134]: 0.7069446363103981

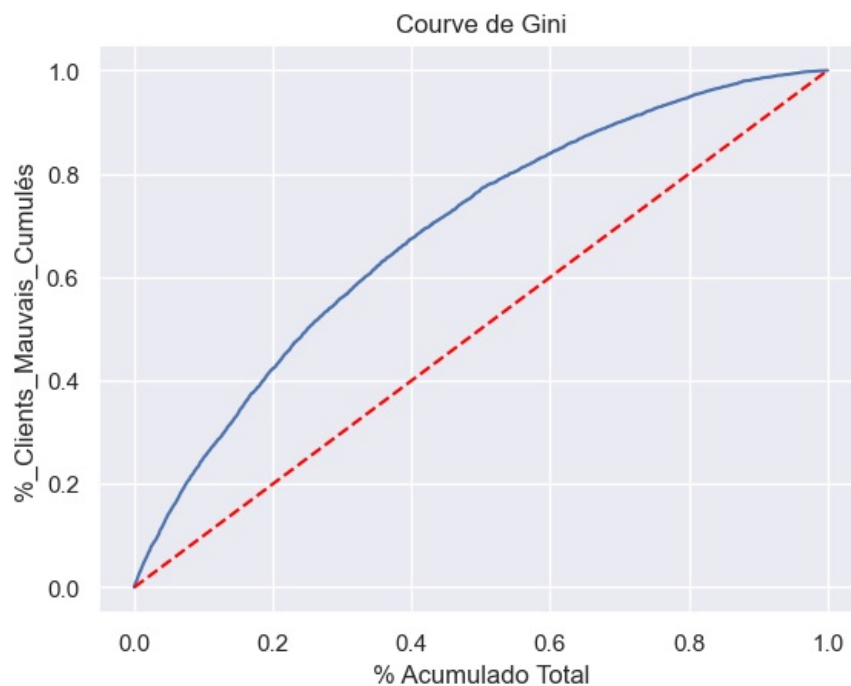
Coefficients de Gini y Kolgomorov-Smirnov

Gini : Mesure de l'inégalité entre les emprunteurs bons et mauvais. Et pourcentage cumulé des mauvais. Axe des X : Cumul total. Plus grande est l'aire sous la courbe, meilleur est le modèle.

Kolmogorov-Smirnov : Mesure à quel point le modèle sépare bien les bons et les mauvais. Plus ils sont séparés, meilleur est le modèle.

```
In [160... plt.plot(df_predict['%_Clients_Cumulés'], df_predict['%_Clients_Mauvais_Cumulés'])
plt.plot(df_predict['%_Clients_Cumulés'], df_predict['%_Clients_Cumulés'], linestyle = '--', color = 'red') # B
plt.xlabel('% Acumulado Total')
plt.ylabel('%_Clients_Mauvais_Cumulés')
plt.title('Courbe de Gini')
```

Out[160]: Text(0.5, 1.0, 'Courbe de Gini')



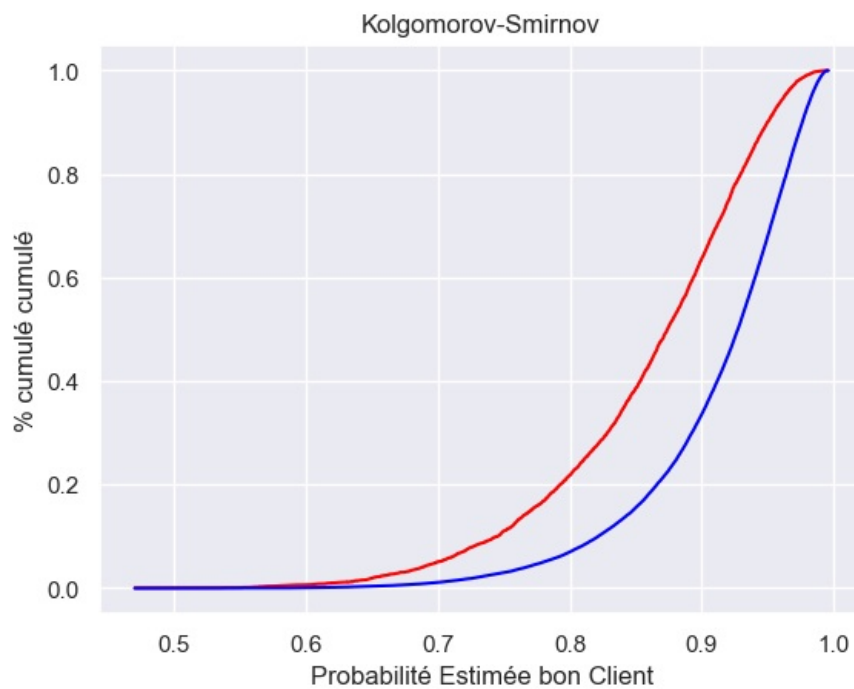
```
In [163... #Un indice de Gini de 0 indiquerait que les emprunteurs à risque sont répartis de manière équitable
#sur l'ensemble de la gamme de notation ; en d'autres termes, le score de crédit n'a pas attribué
#de scores plus bas à davantage d'emprunteurs en défaut, comme le ferait un indicateur avec un pouvoir
#prédictif plus fort. Un indice de coefficient de 100 % indiquerait que tous les emprunteurs en défaut
#se sont vu attribuer avec succès les scores les plus bas possibles. Les coefficients de Gini de 45 %
#ou plus sont considérés comme des indicateurs d'une forte précision dans l'évaluation du crédit.
#Un coefficient de 41,4 % est jugé satisfaisant.
```

```
In [164... # graph K-S
#Rappelons que la statistique de K-S mesure la distance (sur l'axe des Y)
#entre deux fonctions de distribution cumulée. Plus la distance est grande,
#meilleure est la caractéristique qui les distingue. Dans notre cas spécifique, les fonctions sont :

#% Cumulatif de Clients Bons en fonction de la Probabilité d'être bon (Proba_Prediccion)
#% Cumulatif de Clients Mauvais en fonction de la Probabilité d'être bon
#Si le modèle était parfait, la distance maximale serait égale à 1. Pour un modèle de prédiction aléatoire, la
```

```
In [165... plt.plot(df_predict['Proba_Prediccion'], df_predict['% Clients Mauvais_Cumulés'], color = 'red')
plt.plot(df_predict['Proba_Prediccion'], df_predict['% Clients Bons_Cumulés'], color = 'blue')
plt.xlabel('Probabilité Estimée bon Client')
plt.ylabel('% cumulé cumulé')
plt.title('Kolgomorov-Smirnov')
```

```
Out[165]: Text(0.5, 1.0, 'Kolgomorov-Smirnov')
```



```
In [166...] #Le coefficient K-S est la distance (verticale) maximale entre la courbe rouge et la courbe bleue.  
#Nous pouvons le calculer avec les données du dataframe en prenant le maximum de la différence entre  
#le % cumulatif des mauvais et le % cumulatif des bons :
```

```
In [167...] Coef_KS = max(df_predict['%_Clients_Mauvais_Cumulés']-df_predict['%_Clients_Bons_Cumulés'])
```

```
In [168...] Coef_KS
```

```
Out[168]: 0.3045755171087095
```

```
In [169...] #Il ne s'approche pas de 1, mais il est significativement plus élevé que zéro.  
#Les deux distributions cumulatives sont suffisamment éloignées.  
#Nous pouvons affirmer que le modèle a un pouvoir de prédiction satisfaisant.
```

Application et suivi du Modèle PD

Après d'avoir estimé la probabilité de défaut (PD) et de non-défaut (1-PD) à partir de notre modèle de prédiction. nous appliquerons le modèle à partir du calcul d'un score de crédit.

```
In [171...] pd.options.display.max_columns = None
```

```
In [172...] x_test.head()
```

Out[172]:

Unnamed: 0	id	member_id	loan_amnt	funded_amnt	funded_amnt_inv	term	int_rate	installment	grade	sub_grade	e	
89223	89223	7073644	8735123	9600	9600	9600.0	36 months	15.10	333.26	C	C2	J
158835	158835	3640390	4592970	14000	14000	14000.0	36 months	7.62	436.26	A	A3	Caref
108909	108909	6527485	8079529	16750	16750	16750.0	36 months	8.90	531.87	A	A5	Co
270155	270155	32419070	35032306	19750	19750	19750.0	60 months	20.20	525.46	E	E3	
23971	23971	606796	778433	15000	9475	9450.0	60 months	8.88	196.14	B	B1	E Inde

In [173]: tableau

Out[173]:

	Variable Independante	Coefficientes	p_values
0	Intercepto	-0.667983	NaN
1	Home_MORTGAGE	0.099254	4.826906e-15
2	Home_OWN	0.083285	3.966229e-05
3	Adresse_FL	-0.207613	3.308666e-16
4	Adresse_AL_HI_MO_NM	-0.156218	3.667267e-07
5	Adresse_CA	-0.137447	1.484883e-10
6	Adresse_NC_ID_NJ	-0.140432	9.642841e-08
7	Adresse_NY	-0.116864	2.028481e-06
8	Adresse_KY_LA_MD	-0.123486	3.996527e-05
9	Adresse_MI_AR_AZ_VA_OK_DE_OH	-0.079535	3.239793e-04
10	Adresse_MN_PA_UT_MA_RI_WA_TN_IN	-0.075609	5.771834e-04
11	Adresse_CT_IL	0.046172	1.162430e-01
12	Adresse_TX	0.056687	3.165762e-02
13	Adresse_NH_AK_MT_MS_WY_WV_DC_ME	0.280750	1.870772e-10
14	Verification_Not Verified	0.095455	5.203965e-11
15	Verification_Source Verified	-0.005208	7.020393e-01
16	Purpose_house_other_wedding_medical_vacation	0.389900	2.034026e-33
17	Purpose_major_purchase_improvement_car	0.455846	5.272720e-42
18	Purpose_debt_consolidation	0.400467	1.846630e-44
19	Purpose_credit_card	0.531613	1.113748e-66
20	Grade_A	2.008301	0.000000e+00
21	Grade_B	1.323835	0.000000e+00
22	Grade_C	0.876908	8.340840e-291
23	Grade_D	0.553558	1.856190e-116
24	Grade_E	0.259348	3.012767e-24
25	initial_list_status_w	0.063553	2.545807e-06
26	echeance_36	0.034634	1.338126e-02
27	ancianite_1a4	0.054371	3.200932e-04
28	ancianite_5a6	0.037888	9.647531e-02
29	ancianite_7	0.025677	2.999673e-01
30	ancianite_8a9	0.002770	9.174683e-01
31	ancianite_10+	0.097862	1.549211e-11
32	nb_mois_1er_credit_<87	1.813029	4.383749e-148
33	nb_mois_1er_credit_87a89	1.486218	3.580371e-110
34	nb_mois_1er_credit_89a90	1.304625	4.417178e-76

35	nb_mois_1er_credit_90a98	0.992666	2.705136e-55
36	nb_mois_1er_credit_98a101	0.693929	7.641018e-27
37	nb_mois_1er_credit_101a110	0.394082	4.202029e-10
38	nb_mois_1er_credit_110a126	0.072298	2.545465e-01
39	nb_mois_1er_credit_126a155	0.231783	3.324138e-04
40	Revenu_20K-30K	-0.111294	3.967872e-03
41	Revenu_30K-40K	-0.051059	1.484027e-01
42	Revenu_40K-50K	0.020129	5.630304e-01
43	Revenu_50K-60K	0.127166	3.112796e-04
44	Revenu_60K-70K	0.167666	2.762332e-06
45	Revenu_70K-80K	0.273690	1.439970e-13
46	Revenu_80K-90K	0.335496	4.341269e-18
47	Revenu_90K-100K	0.399993	1.689195e-22
48	Revenu_100K-126K	0.480193	1.566924e-35
49	Revenu_126K-152K	0.548754	8.051498e-32
50	Revenu_152K-227K	0.564586	1.691447e-29
51	Revenu_>227K	0.602758	1.113899e-17
52	mths_since_last_delinq_null	0.305900	2.822434e-04
53	mths_since_last_delinq_0-4	0.289052	6.600199e-04
54	mths_since_last_delinq_4-30	0.384151	6.219555e-06
55	mths_since_last_delinq_30-60	0.407221	1.783825e-06
56	mths_since_last_delinq_60-83	0.350782	4.972709e-05
57	total_acc_6-22	-0.063092	9.368739e-02
58	total_acc_22-50	-0.126080	1.237349e-03
59	total_acc_>50	-0.131218	1.009810e-02
60	dti<=3.2	0.261988	2.977315e-11
61	dti_3.2-8.8	0.301200	5.464292e-46
62	dti_8.8-10.4	0.226738	7.638654e-17
63	dti_10.4-13.6	0.187562	5.852196e-22
64	dti_13.6-16.0	0.183184	2.298727e-19
65	dti_16.0-16.7	0.136840	2.773811e-05
66	dti_16.7-19.9	0.105962	3.623664e-09
67	dti_19.9-20.8	0.094623	1.166047e-03
68	dti_20.8-23.2	0.038756	5.353195e-02
69	dti>35.2	-0.091007	1.921946e-01
70	mths_since_last_record_null	0.188450	4.183036e-02
71	mths_since_last_record_3-21	0.312915	1.638101e-02
72	mths_since_last_record_21-31	0.305707	1.605904e-02
73	mths_since_last_record_31-85	0.379676	7.596651e-05
74	mths_since_last_record>85	0.112414	2.366746e-01

Calcul de la PD pour une observation individuelle

Estimons (1-PD) pour la première observation 89223:

On calcul la somme des coefficients pour lesquels le client 89223 a des variables indicatrices avec une valeur de 1, étant donné que les variables avec une valeur de 0 donneront toujours un résultat de 0

intercept -0,667983 +

Home_own (1*0,083285) +

Adresse_CT_IL (1*0,046172)

etc., en multipliant toujours par 1 qui est implicite.

In [174..

```
# Log des ODDS
-0.667983 + (1*0.083285) + (1*0.046172) + (1*0.095455) + 0.531613 + 0.876908 + 0.063553 + 0.034634 + 0.054371 +
```

```
Out[174]: 2.3082059999999998
```

Alors, ici on a estimé le logarithme des cotes : $\ln((1-PD)/PD) = 2,308207$.

Maintenant on applique l'exponentielle des deux côtés pour obtenir $(1-PD)/PD$

```
In [175]: np.exp(2.308207)
```

```
Out[175]: 10.056377395809957
```

```
In [176]: # comme on cherche 1 - PD
# 1 - PD = 10.056377395809957 / (10.056377395809957 + 1)
```

```
In [177]: 10.056377395809957 / (10.056377395809957 + 1)
```

```
Out[177]: 0.9095544621714007
```

```
In [178]: proba_bon
#on verifie qu'on obtient la même proba de non default 0.90955448 qu'auparavant
```

```
Out[178]: array([0.90955448, 0.96180255, 0.9367632 , ..., 0.93635228, 0.88207315,
0.98353305])
```

Score de Crédit

```
In [179]: # On prend les coeff
```

```
In [180]: tableau
```

```
Out[180]:
```

	Variable Independante	Coefficientes	p_values
0	Intercepto	-0.667983	NaN
1	Home_MORTGAGE	0.099254	4.826906e-15
2	Home_OWN	0.083285	3.966229e-05
3	Adresse_FL	-0.207613	3.308666e-16
4	Adresse_AL_HI_MO_NM	-0.156218	3.667267e-07
5	Adresse_CA	-0.137447	1.484883e-10
6	Adresse_NC_ID_NJ	-0.140432	9.642841e-08
7	Adresse_NY	-0.116864	2.028481e-06
8	Adresse_KY_LA_MD	-0.123486	3.996527e-05
9	Adresse_MI_AR_AZ_VA_OK_DE_OH	-0.079535	3.239793e-04
10	Adresse_MN_PA_UT_MA_RI_WA_TN_IN	-0.075609	5.771834e-04
11	Adresse_CT_IL	0.046172	1.162430e-01
12	Adresse_TX	0.056687	3.165762e-02
13	Adresse_NH_AK_MT_MS_WY_WV_DC_ME	0.280750	1.870772e-10
14	Verification_Not Verified	0.095455	5.203965e-11
15	Verification_Source Verified	-0.005208	7.020393e-01
16	Purpose_house_other_wedding_medical_vacation	0.389900	2.034026e-33
17	Purpose_major_purchase_improvement_car	0.455846	5.272720e-42
18	Purpose_debt_consolidation	0.400467	1.846630e-44
19	Purpose_credit_card	0.531613	1.113748e-66
20	Grade_A	2.008301	0.000000e+00
21	Grade_B	1.323835	0.000000e+00
22	Grade_C	0.876908	8.340840e-291
23	Grade_D	0.553558	1.856190e-116
24	Grade_E	0.259348	3.012767e-24
25	initial_list_status_w	0.063553	2.545807e-06
26	echeance_36	0.034634	1.338126e-02
27	ancianite_1a4	0.054371	3.200932e-04
28	ancianite_5a6	0.037888	9.647531e-02
29	ancianite_7	0.025677	2.999673e-01
30	ancianite_8a9	0.002770	9.174683e-01
31	ancianite_10+	0.097862	1.549211e-11
32	nb_mois_1er_credit_<87	1.813029	4.383749e-148

33	nb_mois_1er_credit_87a89	1.486218	3.580371e-110
34	nb_mois_1er_credit_89a90	1.304625	4.417178e-76
35	nb_mois_1er_credit_90a98	0.992666	2.705136e-55
36	nb_mois_1er_credit_98a101	0.693929	7.641018e-27
37	nb_mois_1er_credit_101a110	0.394082	4.202029e-10
38	nb_mois_1er_credit_110a126	0.072298	2.545465e-01
39	nb_mois_1er_credit_126a155	0.231783	3.324138e-04
40	Revenu_20K-30K	-0.111294	3.967872e-03
41	Revenu_30K-40K	-0.051059	1.484027e-01
42	Revenu_40K-50K	0.020129	5.630304e-01
43	Revenu_50K-60K	0.127166	3.112796e-04
44	Revenu_60K-70K	0.167666	2.762332e-06
45	Revenu_70K-80K	0.273690	1.439970e-13
46	Revenu_80K-90K	0.335496	4.341269e-18
47	Revenu_90K-100K	0.399993	1.689195e-22
48	Revenu_100K-126K	0.480193	1.566924e-35
49	Revenu_126K-152K	0.548754	8.051498e-32
50	Revenu_152K-227K	0.564586	1.691447e-29
51	Revenu_>227K	0.602758	1.113899e-17
52	mths_since_last_delinq_null	0.305900	2.822434e-04
53	mths_since_last_delinq_0-4	0.289052	6.600199e-04
54	mths_since_last_delinq_4-30	0.384151	6.219555e-06
55	mths_since_last_delinq_30-60	0.407221	1.783825e-06
56	mths_since_last_delinq_60-83	0.350782	4.972709e-05
57	total_acc_6-22	-0.063092	9.368739e-02
58	total_acc_22-50	-0.126080	1.237349e-03
59	total_acc_>50	-0.131218	1.009810e-02
60	dti<=3.2	0.261988	2.977315e-11
61	dti_3.2-8.8	0.301200	5.464292e-46
62	dti_8.8-10.4	0.226738	7.638654e-17
63	dti_10.4-13.6	0.187562	5.852196e-22
64	dti_13.6-16.0	0.183184	2.298727e-19
65	dti_16.0-16.7	0.136840	2.773811e-05
66	dti_16.7-19.9	0.105962	3.623664e-09
67	dti_19.9-20.8	0.094623	1.166047e-03
68	dti_20.8-23.2	0.038756	5.353195e-02
69	dti>35.2	-0.091007	1.921946e-01
70	mths_since_last_record_null	0.188450	4.183036e-02
71	mths_since_last_record_3-21	0.312915	1.638101e-02
72	mths_since_last_record_21-31	0.305707	1.605904e-02
73	mths_since_last_record_31-85	0.379676	7.596651e-05
74	mths_since_last_record>85	0.112414	2.366746e-01

Lors de la construction du modèle, nous devons exclure une variable de référence avant d'exécuter notre régression logistique. Cependant, dans notre score de notation de crédit, nous devons inclure toutes les variables indicatrices. Par conséquent, nous devons créer un dataframe similaire au tableau de résultats avec les catégories de référence, puis les concaténer au dataframe de notre scorecard.

In [181]:

```
ref
```



```
Out[181]: ['Home_RENT_ANY_OTHER_NONE',
'Adresse_ND_NE_IA_NV',
'Verification_Verified',
'Purpose_ed_pyme_enerren_moving',
'Grades_F_G',
'initial_list_status_f',
'echeance_60',
'ancianite_<_1',
'nb_mois_1er_credit_>155',
'Revenu_<20K',
'mths_since_last_delinq_83+',
'total_acc_<=6',
'dti_23.2-35.2',
'mths_since_last_record_0-3']
```

```
In [182... # Nous créons un dataframe de référence avec les mêmes colonnes que le tableau afin de pouvoir les concaténer.
```

```
In [183... df_ref= pd.DataFrame(ref, columns = ['Variable Independante'])
df_ref['Coeficientes'] = 0
df_ref['p_values'] = np.nan
```

```
In [184... df_ref
```

Out[184]:

	Variable Independante	Coeficientes	p_values
0	Home_RENT_ANY_OTHER_NONE	0	NaN
1	Adresse_ND_NE_IA_NV	0	NaN
2	Verification_Verified	0	NaN
3	Purpose_ed_pyme_enerren_moving	0	NaN
4	Grades_F_G	0	NaN
5	initial_list_status_f	0	NaN
6	echeance_60	0	NaN
7	ancianite_<_1	0	NaN
8	nb_mois_1er_credit_>155	0	NaN
9	Revenu_<20K	0	NaN
10	mths_since_last_delinq_83+	0	NaN
11	total_acc_<=6	0	NaN
12	dti_23.2-35.2	0	NaN
13	mths_since_last_record_0-3	0	NaN

```
In [185... df_scores = pd.concat([tableau, df_ref])
```

```
In [186... df_scores
```

Out[186]:

	Variable Independante	Coeficientes	p_values
0	Intercepto	-0.667983	NaN
1	Home_MORTGAGE	0.099254	4.826906e-15
2	Home_OWN	0.083285	3.966229e-05
3	Adresse_FL	-0.207613	3.308666e-16
4	Adresse_AL_HI_MO_NM	-0.156218	3.667267e-07
5	Adresse_CA	-0.137447	1.484883e-10
6	Adresse_NC_ID_NJ	-0.140432	9.642841e-08
7	Adresse_NY	-0.116864	2.028481e-06
8	Adresse_KY_LA_MD	-0.123486	3.996527e-05
9	Adresse_MI_AR_AZ_VA_OK_DE_OH	-0.079535	3.239793e-04
10	Adresse_MN_PA_UT_MA_RI_WA_TN_IN	-0.075609	5.771834e-04
11	Adresse_CT_IL	0.046172	1.162430e-01
12	Adresse_TX	0.056687	3.165762e-02
13	Adresse_NH_AK_MT_MS_WY_WV_DC_ME	0.280750	1.870772e-10
14	Verification_Not Verified	0.095455	5.203965e-11
15	Verification_Source Verified	-0.005208	7.020393e-01
16	Purpose_house_other_wedding_medical_vacation	0.389900	2.034026e-33
17	Purpose_major_purchase_improvement_car	0.455846	5.272720e-42
18	Purpose_debt_consolidation	0.400467	1.846630e-44
19	Purpose_credit_card	0.531613	1.113748e-66
20	Grade_A	2.008301	0.000000e+00

0	Home_RENT_ANY_OTHER_NONE	0.000000	NaN
1	Adresse_ND_NE_IA_NV	0.000000	NaN
2	Verification_Verified	0.000000	NaN
3	Purpose_ed_pyme_enerren_moving	0.000000	NaN
4	Grades_F_G	0.000000	NaN
5	initial_list_status_f	0.000000	NaN
6	echeance_60	0.000000	NaN
7	ancianite_<_1	0.000000	NaN
8	nb_mois_1er_credit_>155	0.000000	NaN
9	Revenu_<20K	0.000000	NaN
10	mths_since_last_delinq_83+	0.000000	NaN
11	total_acc_<=6	0.000000	NaN
12	dti_23.2-35.2	0.000000	NaN
13	mths_since_last_record_0-3	0.000000	NaN

In [187...

df_scores = df_scores.reset_index()

In [188...

df_scores

Out[188]:

	index	Variable Independante	Coefficientes	p_values
0	0	Intercepto	-0.667983	NaN
1	1	Home_MORTGAGE	0.099254	4.826906e-15
2	2	Home_OWN	0.083285	3.966229e-05
3	3	Adresse_FL	-0.207613	3.308666e-16
4	4	Adresse_AL_HI_MO_NM	-0.156218	3.667267e-07
5	5	Adresse_CA	-0.137447	1.484883e-10
6	6	Adresse_NC_ID_NJ	-0.140432	9.642841e-08
7	7	Adresse_NY	-0.116864	2.028481e-06
8	8	Adresse_KY_LA_MD	-0.123486	3.996527e-05
9	9	Adresse_MI_AR_AZ_VA_OK_DE_OH	-0.079535	3.239793e-04
10	10	Adresse_MN_PA_UT_MA_RI_WA_TN_IN	-0.075609	5.771834e-04
11	11	Adresse_CT_IL	0.046172	1.162430e-01
12	12	Adresse_TX	0.056687	3.165762e-02
13	13	Adresse_NH_AK_MT_MS_WY_WV_DC_ME	0.280750	1.870772e-10
14	14	Verification_Not Verified	0.095455	5.203965e-11
15	15	Verification_Source Verified	-0.005208	7.020393e-01
16	16	Purpose_house_other_wedding_medical_vacation	0.389900	2.034026e-33
17	17	Purpose_major_purchase_improvement_car	0.455846	5.272720e-42
18	18	Purpose_debt_consolidation	0.400467	1.846630e-44
19	19	Purpose_credit_card	0.531613	1.113748e-66
20	20	Grade_A	2.008301	0.000000e+00
21	21	Grade_B	1.323835	0.000000e+00
22	22	Grade_C	0.876908	8.340840e-291
23	23	Grade_D	0.553558	1.856190e-116
24	24	Grade_E	0.259348	3.012767e-24
25	25	initial_list_status_w	0.063553	2.545807e-06
26	26	echeance_36	0.034634	1.338126e-02
27	27	ancianite_1a4	0.054371	3.200932e-04
28	28	ancianite_5a6	0.037888	9.647531e-02
29	29	ancianite_7	0.025677	2.999673e-01
30	30	ancianite_8a9	0.002770	9.174683e-01
31	31	ancianite_10+	0.097862	1.549211e-11
32	32	nb_mois_1er_credit_<87	1.813029	4.383749e-148
33	33	nb_mois_1er_credit_87a89	1.486218	3.580371e-110
34	34	nb_mois_1er_credit_89a90	1.304625	4.417178e-76
35	35	nb_mois_1er_credit_90a98	0.992666	2.705136e-55

```
In [189]: df_scores['Variable Original'] = df_scores['Variable Independante'].str.split('_').str[0]
```

```
In [190]: df_scores
```

Out[190]:

	index	Variable Independante	Coefficientes	p_values	Variable Original
0	0	Intercepto	-0.667983	NaN	Intercepto
1	1	Home_MORTGAGE	0.099254	4.826906e-15	Home
2	2	Home_OWN	0.083285	3.966229e-05	Home
3	3	Adresse_FL	-0.207613	3.308666e-16	Adresse
4	4	Adresse_AL_HI_MO_NM	-0.156218	3.667267e-07	Adresse
5	5	Adresse_CA	-0.137447	1.484883e-10	Adresse
6	6	Adresse_NC_ID_NJ	-0.140432	9.642841e-08	Adresse
7	7	Adresse_NY	-0.116864	2.028481e-06	Adresse
8	8	Adresse_KY_LA_MD	-0.123486	3.996527e-05	Adresse
9	9	Adresse_MI_AR_AZ_VA_OK_DE_OH	-0.079535	3.239793e-04	Adresse
10	10	Adresse_MN_PA_UT_MA_RI_WA_TN_IN	-0.075609	5.771834e-04	Adresse
11	11	Adresse_CT_IL	0.046172	1.162430e-01	Adresse
12	12	Adresse_TX	0.056687	3.165762e-02	Adresse
13	13	Adresse_NH_AK_MT_MS_WY_WV_DC_ME	0.280750	1.870772e-10	Adresse
14	14	Verification_Not Verified	0.095455	5.203965e-11	Verification
15	15	Verification_Source Verified	-0.005208	7.020393e-01	Verification
16	16	Purpose_house_other_wedding_medical_vacation	0.389900	2.034026e-33	Purpose
17	17	Purpose_major_purchase_improvement_car	0.455846	5.272720e-42	Purpose
18	18	Purpose_debt_consolidation	0.400467	1.846630e-44	Purpose
19	19	Purpose_credit_card	0.531613	1.113748e-66	Purpose
20	20	Grade_A	2.008301	0.000000e+00	Grade
21	21	Grade_B	1.323835	0.000000e+00	Grade
22	22	Grade_C	0.876908	8.340840e-291	Grade
23	23	Grade_D	0.553558	1.856190e-116	Grade
24	24	Grade_E	0.259348	3.012767e-24	Grade
25	25	initial_list_status_w	0.063553	2.545807e-06	initial
26	26	echeance_36	0.034634	1.338126e-02	echeance
27	27	ancianite_1a4	0.054371	3.200932e-04	ancianite
28	28	ancianite_5a6	0.037888	9.647531e-02	ancianite
29	29	ancianite_7	0.025677	2.999673e-01	ancianite
30	30	ancianite_8a9	0.002770	9.174683e-01	ancianite
31	31	ancianite_10+	0.097862	1.549211e-11	ancianite
32	32	nb_mois_1er_credit_<87	1.813029	4.383749e-148	nb
33	33	nb_mois_1er_credit_87a89	1.486218	3.580371e-110	nb
34	34	nb_mois_1er_credit_89a90	1.304625	4.417178e-76	nb
35	35	nb_mois_1er_credit_90a98	0.992666	2.705136e-55	nb
36	36	nb_mois_1er_credit_98a101	0.693929	7.641018e-27	nb
37	37	nb_mois_1er_credit_101a110	0.394082	4.202029e-10	nb
38	38	nb_mois_1er_credit_110a126	0.072298	2.545465e-01	nb
39	39	nb_mois_1er_credit_126a155	0.231783	3.324138e-04	nb
40	40	Revenu_20K-30K	-0.111294	3.967872e-03	Revenu
41	41	Revenu_30K-40K	-0.051059	1.484027e-01	Revenu
42	42	Revenu_40K-50K	0.020129	5.630304e-01	Revenu
43	43	Revenu_50K-60K	0.127166	3.112796e-04	Revenu
44	44	Revenu_60K-70K	0.167666	2.762332e-06	Revenu
45	45	Revenu_70K-80K	0.273690	1.439970e-13	Revenu
46	46	Revenu_80K-90K	0.335496	4.341269e-18	Revenu
47	47	Revenu_90K-100K	0.399993	1.689195e-22	Revenu
48	48	Revenu_100K-126K	0.480193	1.566924e-35	Revenu
49	49	Revenu_126K-152K	0.548754	8.051498e-32	Revenu
50	50	Revenu_152K-227K	0.564586	1.691447e-29	Revenu

51	51	Revenu_>227K	0.602758	1.113899e-17	Revenu
52	52	mths_since_last_delinq_null	0.305900	2.822434e-04	mths
53	53	mths_since_last_delinq_0-4	0.289052	6.600199e-04	mths
54	54	mths_since_last_delinq_4-30	0.384151	6.219555e-06	mths
55	55	mths_since_last_delinq_30-60	0.407221	1.783825e-06	mths
56	56	mths_since_last_delinq_60-83	0.350782	4.972709e-05	mths
57	57	total_acc_6-22	-0.063092	9.368739e-02	total
58	58	total_acc_22-50	-0.126080	1.237349e-03	total
59	59	total_acc_>50	-0.131218	1.009810e-02	total
60	60	dti<=3.2	0.261988	2.977315e-11	dti<=3.2
61	61	dti_3.2-8.8	0.301200	5.464292e-46	dti
62	62	dti_8.8-10.4	0.226738	7.638654e-17	dti
63	63	dti_10.4-13.6	0.187562	5.852196e-22	dti
64	64	dti_13.6-16.0	0.183184	2.298727e-19	dti
65	65	dti_16.0-16.7	0.136840	2.773811e-05	dti
66	66	dti_16.7-19.9	0.105962	3.623664e-09	dti
67	67	dti_19.9-20.8	0.094623	1.166047e-03	dti
68	68	dti_20.8-23.2	0.038756	5.353195e-02	dti
69	69	dti>35.2	-0.091007	1.921946e-01	dti>35.2
70	70	mths_since_last_record_null	0.188450	4.183036e-02	mths
71	71	mths_since_last_record_3-21	0.312915	1.638101e-02	mths
72	72	mths_since_last_record_21-31	0.305707	1.605904e-02	mths
73	73	mths_since_last_record_31-85	0.379676	7.596651e-05	mths
74	74	mths_since_last_record>85	0.112414	2.366746e-01	mths
75	0	Home_RENT_ANY_OTHER_NONE	0.000000	NaN	Home
76	1	Adresse_ND_NE_IA_NV	0.000000	NaN	Adresse
77	2	Verification_Verified	0.000000	NaN	Verification
78	3	Purpose_ed_pyme_enerren_moving	0.000000	NaN	Purpose
79	4	Grades_F_G	0.000000	NaN	Grades
80	5	initial_list_status_f	0.000000	NaN	initial
81	6	echeance_60	0.000000	NaN	echeance
82	7	ancianite_<_1	0.000000	NaN	ancianite
83	8	nb_mois_1er_credit_>155	0.000000	NaN	nb
84	9	Revenu_<20K	0.000000	NaN	Revenu
85	10	mths_since_last_delinq_83+	0.000000	NaN	mths
86	11	total_acc_<=6	0.000000	NaN	total
87	12	dti_23.2-35.2	0.000000	NaN	dti
88	13	mths_since_last_record_0-3	0.000000	NaN	mths

```
In [191]: #Nous rencontrons quelques petits problèmes avec dti<=3.2 et dti>35.2
#on corrige corrige les variables ayant des noms comme "mois" ou "mths".
```

```
In [192]: df_scores.loc[60, 'Variable Original'] = 'dti'
df_scores.loc[69, 'Variable Original'] = 'dti'
df_scores.loc[32:39, 'Variable Original'] = 'mois_credit'
df_scores.loc[83, 'Variable Original'] = 'mois_credit'
df_scores.loc[52:56, 'Variable Original'] = 'mths_since_last_delinq'
df_scores.loc[85, 'Variable Original'] = 'mths_since_last_delinq'
df_scores.loc[70:74, 'Variable Original'] = 'mths_since_last_record'
df_scores.loc[88, 'Variable Original'] = 'mths_since_last_record'
df_scores.loc[57:59, 'Variable Original'] = 'total_acc'
df_scores.loc[86, 'Variable Original'] = 'total_acc'
```

```
In [193]: df_scores
```

```
Out[193]:
```

	index	Variable Independante	Coefficientes	p_values	Variable Original
0	0	Intercepto	-0.667983	NaN	Intercepto
1	1	Home_MORTGAGE	0.099254	4.826906e-15	Home
2	2	Home_OWN	0.083285	3.966229e-05	Home
3	3	Adresse_FL	-0.207613	3.308666e-16	Adresse

59	59	total_acc_>50	-0.131218	1.009810e-02	total_acc
60	60	dti<=3.2	0.261988	2.977315e-11	dti
61	61	dti_3.2-8.8	0.301200	5.464292e-46	dti
62	62	dti_8.8-10.4	0.226738	7.638654e-17	dti
63	63	dti_10.4-13.6	0.187562	5.852196e-22	dti
64	64	dti_13.6-16.0	0.183184	2.298727e-19	dti
65	65	dti_16.0-16.7	0.136840	2.773811e-05	dti
66	66	dti_16.7-19.9	0.105962	3.623664e-09	dti
67	67	dti_19.9-20.8	0.094623	1.166047e-03	dti
68	68	dti_20.8-23.2	0.038756	5.353195e-02	dti
69	69	dti>35.2	-0.091007	1.921946e-01	dti
70	70	mths_since_last_record_null	0.188450	4.183036e-02	mths_since_last_record
71	71	mths_since_last_record_3-21	0.312915	1.638101e-02	mths_since_last_record
72	72	mths_since_last_record_21-31	0.305707	1.605904e-02	mths_since_last_record
73	73	mths_since_last_record_31-85	0.379676	7.596651e-05	mths_since_last_record
74	74	mths_since_last_record>85	0.112414	2.366746e-01	mths_since_last_record
75	0	Home_RENT_ANY_OTHER_NONE	0.000000	NaN	Home
76	1	Adresse_ND_NE_IA_NV	0.000000	NaN	Adresse
77	2	Verification_Verified	0.000000	NaN	Verification
78	3	Purpose_ed_pyme_enerren_moving	0.000000	NaN	Purpose
79	4	Grades_F_G	0.000000	NaN	Grades
80	5	initial_list_status_f	0.000000	NaN	initial
81	6	echeance_60	0.000000	NaN	echeance
82	7	ancianite_<_1	0.000000	NaN	ancianite
83	8	nb_mois_1er_credit_>155	0.000000	NaN	mois_credit
84	9	Revenu_<20K	0.000000	NaN	Revenu
85	10	mths_since_last_delinq_83+	0.000000	NaN	mths_since_last_delinq
86	11	total_acc_<=6	0.000000	NaN	total_acc
87	12	dti_23.2-35.2	0.000000	NaN	dti
88	13	mths_since_last_record_0-3	0.000000	NaN	mths_since_last_record

Nous devons convertir les coefficients de la régression en scores. Tout d'abord, nous devons décider du score minimum et maximum. Nous nous appuyons sur FICO, un fournisseur de scores de crédit, qui utilise une plage de 300 à 850.

```
In [194... score_min = 300
score_max = 850
```

```
In [195... #En plus des valeurs min-max du score, nous avons besoin de l'évaluation de crédit qui peut être obtenue à part.
#modèle. Rappelons-nous quand nous avons fait le calcul pour une observation individuelle, chaque observation t
#seule catégorie d'indicateurs pour chaque variable d'origine (si c'est une maison propre, ce n'est pas une mai
#coefficients les plus élevés correspondent aux observations les plus susceptibles d'être de bons clients. Par
#score minimum devrait être attribué à une observation qui hypothétiquement tomberait dans les catégories avec
#coefficients. Le contraire s'appliquerait pour le score maximum. Trouvons ces minimums et maximums.
```

```
In [196... #Trouvons la valeur minimale du coefficient pour chaque groupe de variables indicatrices liées à chaque variabl
#Nous regroupons les 'Variables Originales' en fonction de leurs Coefficients et sélectionnons le minimum.
```

```
In [197... df_scores.groupby('Variable Original')['Coeficientes'].min()
```

```
Out[197]: Variable Original
Adresse -0.207613
Grade 0.259348
Grades 0.000000
Home 0.000000
Intercepto -0.667983
Purpose 0.000000
Revenu -0.111294
Verification -0.005208
ancianite 0.000000
dti -0.091007
echeance 0.000000
initial 0.000000
mois_credit 0.000000
mths_since_last_delinq 0.000000
mths_since_last_record 0.000000
total_acc -0.131218
Name: Coeficientes, dtype: float64

In [198... #Dans un monde parfait, le coefficient minimum de chaque variable devrait être zéro. Cependant, rappelons-nous,
#avec des données réelles. En ignorant l'Intercept, nous avons 5 coefficients minimums qui sont
#négatifs. Par conséquent, nous ne pouvons pas assigner 350 au coefficient zéro. Au lieu de cela, nous pouvons
#coefficients minimums.
```

```
In [199... somme_coeficiente_min = df_scores.groupby('Variable Original')['Coeficientes'].min().sum()
```

```
In [200... somme_coeficiente_min
```

Out[200]: -0.9549751858468026

```
In [201... # Analogiquement, estimons les coefficients maximaux.
```

```
In [202... df_scores.groupby('Variable Original')['Coeficientes'].max()
```

```
Out[202]: Variable Original
Adresse 0.280750
Grade 2.008301
Grades 0.000000
Home 0.099254
Intercepto -0.667983
Purpose 0.531613
Revenu 0.602758
Verification 0.095455
ancianite 0.097862
dti 0.301200
echeance 0.034634
initial 0.063553
mois_credit 1.813029
mths_since_last_delinq 0.407221
mths_since_last_record 0.379676
total_acc 0.000000
Name: Coeficientes, dtype: float64
```

```
In [203... somme_coeficiente_max = df_scores.groupby('Variable Original')['Coeficientes'].max().sum()
```

```
In [204... somme_coeficiente_max
```

Out[204]: 6.047322470807669

```
In [205... # reescalemos...
# puntuación = coeficiente * (puntaje_max - puntaje_min) / (suma_coeficiente_max - suma_coeficiente_min)
```

```
In [206... df_scores['Calcul score'] = df_scores['Coeficientes'] * (score_max - score_min) / (somme_coeficiente_max - somm
```

```
In [207... df_scores
```

Out[207]:		index	Variable Independante	Coeficientes	p_values	Variable Original	Calcul score
0	0		Intercepto	-0.667983	NaN	Intercepto	-52.467134
1	1		Home_MORTGAGE	0.099254	4.826906e-15	Home	7.795976
2	2		Home_OWN	0.083285	3.966229e-05	Home	6.541680
3	3		Adresse_FL	-0.207613	3.308666e-16	Adresse	-16.307117
4	4		Adresse_AL_HI_MO_NM	-0.156218	3.667267e-07	Adresse	-12.270275
5	5		Adresse_CA	-0.137447	1.484883e-10	Adresse	-10.795843
6	6		Adresse_NC_ID_NJ	-0.140432	9.642841e-08	Adresse	-11.030333
7	7		Adresse_NY	-0.116864	2.028481e-06	Adresse	-9.179141
8	8		Adresse_KY_LA_MD	-0.123486	3.996527e-05	Adresse	-9.699258
9	9		Adresse_MI_AR_AZ_VA_OK_DE_OH	-0.079535	3.239793e-04	Adresse	-6.247130
10	10		Adresse_MN_PA_UT_MA_RI_WA_TN_IN	-0.075609	5.771834e-04	Adresse	-5.938794

66	66	dti_16.7-19.9	0.105962	3.623664e-09	dti	8.322890
67	67	dti_19.9-20.8	0.094623	1.166047e-03	dti	7.432207
68	68	dti_20.8-23.2	0.038756	5.353195e-02	dti	3.044106
69	69	dti>35.2	-0.091007	1.921946e-01	dti	-7.148186
70	70	mths_since_last_record_null	0.188450	4.183036e-02	mths_since_last_record	14.801942
71	71	mths_since_last_record_3-21	0.312915	1.638101e-02	mths_since_last_record	24.578132
72	72	mths_since_last_record_21-31	0.305707	1.605904e-02	mths_since_last_record	24.011983
73	73	mths_since_last_record_31-85	0.379676	7.596651e-05	mths_since_last_record	29.821887
74	74	mths_since_last_record>85	0.112414	2.366746e-01	mths_since_last_record	8.829646
75	0	Home_RENT_ANY_OTHER_NONE	0.000000	NaN	Home	0.000000
76	1	Adresse_ND_NE_IA_NV	0.000000	NaN	Adresse	0.000000
77	2	Verification_Verified	0.000000	NaN	Verification	0.000000
78	3	Purpose_ed_pyme_enerren_moving	0.000000	NaN	Purpose	0.000000
79	4	Grades_F_G	0.000000	NaN	Grades	0.000000
80	5	initial_list_status_f	0.000000	NaN	initial	0.000000
81	6	echeance_60	0.000000	NaN	echeance	0.000000
82	7	ancianite_<_1	0.000000	NaN	ancianite	0.000000
83	8	nb_mois_1er_credit_>155	0.000000	NaN	mois_credit	0.000000
84	9	Revenu_<20K	0.000000	NaN	Revenu	0.000000
85	10	mths_since_last_delinq_83+	0.000000	NaN	mths_since_last_delinq	0.000000
86	11	total_acc_<=6	0.000000	NaN	total_acc	0.000000
87	12	dti_23.2-35.2	0.000000	NaN	dti	0.000000
88	13	mths_since_last_record_0-3	0.000000	NaN	mths_since_last_record	0.000000

```
In [208.. #Notre intercept est négatif. Nous devons l'ajuster pour qu'il se rapproche de 300, la valeur de score minimum
#définie.
# Intercept_ajusté = (Intercepto_coef - somme_coeficiente_min) * (score_max - score_min) / (somme_coeficiente_m
```

```
In [209.. df_scores.loc[0,'Calcul score'] = (df_scores.loc[0, 'Coeficientes'] - somme_coeficiente_min) * (score_max - sco
```

```
In [210.. df_scores
```

Out[210]:	index	Variable Independante	Coeficientes	p_values	Variable Original	Calcul score
	0	Intercepto	-0.667983	NaN	Intercepto	322.542009
	1	Home_MORTGAGE	0.099254	4.826906e-15	Home	7.795976
	2	Home_OWN	0.083285	3.966229e-05	Home	6.541680
	3	Adresse_FL	-0.207613	3.308666e-16	Adresse	-16.307117
	4	Adresse_AL_HI_MO_NM	-0.156218	3.667267e-07	Adresse	-12.270275
	5	Adresse_CA	-0.137447	1.484883e-10	Adresse	-10.795843
	6	Adresse_NC_ID_NJ	-0.140432	9.642841e-08	Adresse	-11.030333
	7	Adresse_NY	-0.116864	2.028481e-06	Adresse	-9.179141
	8	Adresse_KY_LA_MD	-0.123486	3.996527e-05	Adresse	-9.699258
	9	Adresse_MI_AR_AZ_VA_OK_DE_OH	-0.079535	3.239793e-04	Adresse	-6.247130
	10	Adresse_MN_PA_UT_MA_RI_WA_TN_IN	-0.075609	5.771834e-04	Adresse	-5.938794
	11	Adresse_CT_IL	0.046172	1.162430e-01	Adresse	3.626584
	12	Adresse_TX	0.056687	3.165762e-02	Adresse	4.452541
	13	Adresse_NH_AK_MT_MS_WY_WV_DC_ME	0.280750	1.870772e-10	Adresse	22.051728
	14	Verification_Not Verified	0.095455	5.203965e-11	Verification	7.497551
	15	Verification_Source Verified	-0.005208	7.020393e-01	Verification	-0.409071
	16	Purpose_house_other_wedding_medical_vacation	0.389900	2.034026e-33	Purpose	30.624924
	17	Purpose_major_purchase_improvement_car	0.455846	5.272720e-42	Purpose	35.804741
	18	Purpose_debt_consolidation	0.400467	1.846630e-44	Purpose	31.454971
	19	Purpose_credit_card	0.531613	1.113748e-66	Purpose	41.755855
	20	Grade_A	2.008301	0.000000e+00	Grade	157.743314
	21	Grade_B	1.323835	0.000000e+00	Grade	103.981454
	22	Grade_C	0.876908	8.340840e-291	Grade	68.877330
	23	Grade_D	0.553558	1.856190e-116	Grade	43.479607
	24	Grade_E	0.250348	3.012767e-24	Grade	20.370638

79	4	Grades_F_G	0.000000	NaN	Grades	0.000000
80	5	initial_list_status_f	0.000000	NaN	initial	0.000000
81	6	echeance_60	0.000000	NaN	echeance	0.000000
82	7	ancianite_<_1	0.000000	NaN	ancianite	0.000000
83	8	nb_mois_1er_credit_>155	0.000000	NaN	mois_credit	0.000000
84	9	Revenu_<20K	0.000000	NaN	Revenu	0.000000
85	10	mths_since_last_delinq_83+	0.000000	NaN	mths_since_last_delinq	0.000000
86	11	total_acc_<=6	0.000000	NaN	total_acc	0.000000
87	12	dti_23.2-35.2	0.000000	NaN	dti	0.000000
88	13	mths_since_last_record_0-3	0.000000	NaN	mths_since_last_record	0.000000

In [212..

```
df_scores['Score Round'] = df_scores['Calcul score'].round()
```

In [213..

```
df_scores
#Le fait de vivre dans sa propre maison donne 7 points par exemple
```

Out[213]:

	index	Variable Independante	Coefficientes	p_values	Variable Original	Calcul score	Score Round
	0	Intercepto	-0.667983	NaN	Intercepto	322.542009	323.0
	1	Home_MORTGAGE	0.099254	4.826906e-15	Home	7.795976	8.0
	2	Home_OWN	0.083285	3.966229e-05	Home	6.541680	7.0
	3	Adresse_FL	-0.207613	3.308666e-16	Adresse	-16.307117	-16.0
	4	Adresse_AL_HI_MO_NM	-0.156218	3.667267e-07	Adresse	-12.270275	-12.0
	5	Adresse_CA	-0.137447	1.484883e-10	Adresse	-10.795843	-11.0
	6	Adresse_NC_ID_NJ	-0.140432	9.642841e-08	Adresse	-11.030333	-11.0
	7	Adresse_NY	-0.116864	2.028481e-06	Adresse	-9.179141	-9.0
	8	Adresse_KY_LA_MD	-0.123486	3.996527e-05	Adresse	-9.699258	-10.0
	9	Adresse_MI_AR_AZ_VA_OK_DE_OH	-0.079535	3.239793e-04	Adresse	-6.247130	-6.0
	10	Adresse_MN_PA_UT_MA_RI_WA_TN_IN	-0.075609	5.771834e-04	Adresse	-5.938794	-6.0
	11	Adresse_CT_IL	0.046172	1.162430e-01	Adresse	3.626584	4.0
	12	Adresse_TX	0.056687	3.165762e-02	Adresse	4.452541	4.0
	13	Adresse_NH_AK_MT_MS_WY_WV_DC_ME	0.280750	1.870772e-10	Adresse	22.051728	22.0
	14	Verification_Not Verified	0.095455	5.203965e-11	Verification	7.497551	7.0
	15	Verification_Source Verified	-0.005208	7.020393e-01	Verification	-0.409071	-0.0
	16	Purpose_house_other_wedding_medical_vacation	0.389900	2.034026e-33	Purpose	30.624924	31.0
	17	Purpose_major_purchase_improvement_car	0.455846	5.272720e-42	Purpose	35.804741	36.0
	18	Purpose_debt_consolidation	0.400467	1.846630e-44	Purpose	31.454971	31.0
	19	Purpose_credit_card	0.531613	1.113748e-66	Purpose	41.755855	42.0
	20	Grade_A	2.008301	0.000000e+00	Grade	157.743314	158.0
	21	Grade_B	1.323835	0.000000e+00	Grade	103.981454	104.0
	22	Grade_C	0.876908	8.340840e-291	Grade	68.877330	69.0
	23	Grade_D	0.553558	1.856190e-116	Grade	43.479607	43.0
	24	Grade_E	0.259348	3.012767e-24	Grade	20.370638	20.0
	25	initial_list_status_w	0.063553	2.545807e-06	initial	4.991815	5.0
	26	echeance_36	0.034634	1.338126e-02	echeance	2.720347	3.0
	27	ancianite_1a4	0.054371	3.200932e-04	ancianite	4.270596	4.0
	28	ancianite_5a6	0.037888	9.647531e-02	ancianite	2.975965	3.0
	29	ancianite_7	0.025677	2.999673e-01	ancianite	2.016813	2.0
	30	ancianite_8a9	0.002770	9.174683e-01	ancianite	0.217609	0.0
	31	ancianite_10+	0.097862	1.549211e-11	ancianite	7.686666	8.0
	32	nb_mois_1er_credit_<87	1.813029	4.383749e-148	mois_credit	142.405533	142.0
	33	nb_mois_1er_credit_87a89	1.486218	3.580371e-110	mois_credit	116.735933	117.0
	34	nb_mois_1er_credit_89a90	1.304625	4.417178e-76	mois_credit	102.472620	102.0
	35	nb_mois_1er_credit_90a98	0.992666	2.705136e-55	mois_credit	77.969614	78.0
	36	nb_mois_1er_credit_98a101	0.693929	7.641018e-27	mois_credit	54.505140	55.0
	37	nb_mois_1er_credit_101a110	0.394082	4.202029e-10	mois_credit	30.953398	31.0
	38	nb_mois_1er_credit_110a126	0.072298	2.545465e-01	mois_credit	5.678658	6.0
	39	nb mois 1er credit 126a155	0.231783	3.324138e-04	mois credit	18.205567	18.0

40	40	Revenu_20K-30K	-0.111294	3.967872e-03	Revenu	-8.741648	-9.0
41	41	Revenu_30K-40K	-0.051059	1.484027e-01	Revenu	-4.010460	-4.0
42	42	Revenu_40K-50K	0.020129	5.630304e-01	Revenu	1.581073	2.0
43	43	Revenu_50K-60K	0.127166	3.112796e-04	Revenu	9.988298	10.0
44	44	Revenu_60K-70K	0.167666	2.762332e-06	Revenu	13.169451	13.0
45	45	Revenu_70K-80K	0.273690	1.439970e-13	Revenu	21.497166	21.0
46	46	Revenu_80K-90K	0.335496	4.341269e-18	Revenu	26.351731	26.0
47	47	Revenu_90K-100K	0.399993	1.689195e-22	Revenu	31.417698	31.0
48	48	Revenu_100K-126K	0.480193	1.566924e-35	Revenu	37.717053	38.0
49	49	Revenu_126K-152K	0.548754	8.051498e-32	Revenu	43.102235	43.0
50	50	Revenu_152K-227K	0.564586	1.691447e-29	Revenu	44.345777	44.0
51	51	Revenu_>227K	0.602758	1.113899e-17	Revenu	47.343982	47.0
52	52	mths_since_last_delinq_null	0.305900	2.822434e-04	mths_since_last_delinq	24.027111	24.0
53	53	mths_since_last_delinq_0-4	0.289052	6.600199e-04	mths_since_last_delinq	22.703738	23.0
54	54	mths_since_last_delinq_4-30	0.384151	6.219555e-06	mths_since_last_delinq	30.173383	30.0
55	55	mths_since_last_delinq_30-60	0.407221	1.783825e-06	mths_since_last_delinq	31.985398	32.0
56	56	mths_since_last_delinq_60-83	0.350782	4.972709e-05	mths_since_last_delinq	27.552421	28.0
57	57	total_acc_6-22	-0.063092	9.368739e-02	total_acc	-4.955568	-5.0
58	58	total_acc_22-50	-0.126080	1.237349e-03	total_acc	-9.903019	-10.0
59	59	total_acc_>50	-0.131218	1.009810e-02	total_acc	-10.306625	-10.0
60	60	dti<=3.2	0.261988	2.977315e-11	dti	20.578017	21.0
61	61	dti_3.2-8.8	0.301200	5.464292e-46	dti	23.657938	24.0
62	62	dti_8.8-10.4	0.226738	7.638654e-17	dti	17.809319	18.0
63	63	dti_10.4-13.6	0.187562	5.852196e-22	dti	14.732190	15.0
64	64	dti_13.6-16.0	0.183184	2.298727e-19	dti	14.388279	14.0
65	65	dti_16.0-16.7	0.136840	2.773811e-05	dti	10.748175	11.0
66	66	dti_16.7-19.9	0.105962	3.623664e-09	dti	8.322890	8.0
67	67	dti_19.9-20.8	0.094623	1.166047e-03	dti	7.432207	7.0
68	68	dti_20.8-23.2	0.038756	5.353195e-02	dti	3.044106	3.0
69	69	dti>35.2	-0.091007	1.921946e-01	dti	-7.148186	-7.0
70	70	mths_since_last_record_null	0.188450	4.183036e-02	mths_since_last_record	14.801942	15.0
71	71	mths_since_last_record_3-21	0.312915	1.638101e-02	mths_since_last_record	24.578132	25.0
72	72	mths_since_last_record_21-31	0.305707	1.605904e-02	mths_since_last_record	24.011983	24.0
73	73	mths_since_last_record_31-85	0.379676	7.596651e-05	mths_since_last_record	29.821887	30.0
74	74	mths_since_last_record>85	0.112414	2.366746e-01	mths_since_last_record	8.829646	9.0
75	0	Home_RENT_ANY_OTHER_NONE	0.000000	NaN	Home	0.000000	0.0
76	1	Adresse_ND_NE_IA_NV	0.000000	NaN	Adresse	0.000000	0.0
77	2	Verification_Verified	0.000000	NaN	Verification	0.000000	0.0
78	3	Purpose_ed_pyme_enerren_moving	0.000000	NaN	Purpose	0.000000	0.0
79	4	Grades_F_G	0.000000	NaN	Grades	0.000000	0.0
80	5	initial_list_status_f	0.000000	NaN	initial	0.000000	0.0
81	6	echeance_60	0.000000	NaN	echeance	0.000000	0.0
82	7	ancianite_<_1	0.000000	NaN	ancianite	0.000000	0.0
83	8	nb_mois_1er_credit_>155	0.000000	NaN	mois_credit	0.000000	0.0
84	9	Revenu_<20K	0.000000	NaN	Revenu	0.000000	0.0
85	10	mths_since_last_delinq_83+	0.000000	NaN	mths_since_last_delinq	0.000000	0.0
86	11	total_acc_<=6	0.000000	NaN	total_acc	0.000000	0.0
87	12	dti_23.2-35.2	0.000000	NaN	dti	0.000000	0.0
88	13	mths_since_last_record_0-3	0.000000	NaN	mths_since_last_record	0.000000	0.0

In [214..] *#vérifions si les scores minimums et maximums calculés correspondent à ceux établis.*

In [215..] `score_min_initial = df_scores.groupby('Variable Original')['Score Round'].min().sum()`

In [216..] `score_min_initial`

Out[216]: 301.0

In [217... #Faisons de même pour le score maximum

In [218... score_max_initial = df_scores.groupby('Variable Original')['Score Round'].max().sum()

In [219... score_max_initial

Out[219]: 851.0

In [220... # Il est courant que le score maximum ne corresponde pas exactement, dépassant parfois de un (851) en raison de
Dans ces cas, choisissez la variable avec l'arrondi le plus élevé et soustrayez-en 1

In [221... df_scores['Difference_in_round'] = df_scores['Score Round']- df_scores['Calcul score']

In [222... df_scores

Out[222]:

	index	Variable Independante	Coefficientes	p_values	Variable Original	Calcul score	Score Round	Difference_in
	0	Intercepto	-0.667983	NaN	Intercepto	322.542009	323.0	0.
1	1	Home_MORTGAGE	0.099254	4.826906e-15	Home	7.795976	8.0	0.
2	2	Home_OWN	0.083285	3.966229e-05	Home	6.541680	7.0	0.
3	3	Adresse_FL	-0.207613	3.308666e-16	Adresse	-16.307117	-16.0	0.
4	4	Adresse_AL_HI_MO_NM	-0.156218	3.667267e-07	Adresse	-12.270275	-12.0	0.
5	5	Adresse_CA	-0.137447	1.484883e-10	Adresse	-10.795843	-11.0	-0.
6	6	Adresse_NC_ID_NJ	-0.140432	9.642841e-08	Adresse	-11.030333	-11.0	0.
7	7	Adresse_NY	-0.116864	2.028481e-06	Adresse	-9.179141	-9.0	0.
8	8	Adresse_KY_LA_MD	-0.123486	3.996527e-05	Adresse	-9.699258	-10.0	-0.
9	9	Adresse_MI_AR_AZ_VA_OK_DE_OH	-0.079535	3.239793e-04	Adresse	-6.247130	-6.0	0.
10	10	Adresse_MN_PA_UT_MA_RI_WA_TN_IN	-0.075609	5.771834e-04	Adresse	-5.938794	-6.0	-0.
11	11	Adresse_CT_IL	0.046172	1.162430e-01	Adresse	3.626584	4.0	0.
12	12	Adresse_TX	0.056687	3.165762e-02	Adresse	4.452541	4.0	-0.
13	13	Adresse_NH_AK_MT_MS_WY_WV_DC_ME	0.280750	1.870772e-10	Adresse	22.051728	22.0	-0.
14	14	Verification_Not Verified	0.095455	5.203965e-11	Verification	7.497551	7.0	-0.
15	15	Verification_Source Verified	-0.005208	7.020393e-01	Verification	-0.409071	-0.0	0.
16	16	Purpose_house_other_wedding_medical_vacation	0.389900	2.034026e-33	Purpose	30.624924	31.0	0.
17	17	Purpose_major_purchase_improvement_car	0.455846	5.272720e-42	Purpose	35.804741	36.0	0.
18	18	Purpose_debt_consolidation	0.400467	1.846630e-44	Purpose	31.454971	31.0	-0.
19	19	Purpose_credit_card	0.531613	1.113748e-66	Purpose	41.755855	42.0	0.
20	20	Grade_A	2.008301	0.000000e+00	Grade	157.743314	158.0	0.
21	21	Grade_B	1.323835	0.000000e+00	Grade	103.981454	104.0	0.
22	22	Grade_C	0.876908	8.340840e-291	Grade	68.877330	69.0	0.
23	23	Grade_D	0.553558	1.856190e-116	Grade	43.479607	43.0	-0.
24	24	Grade_E	0.259348	3.012767e-24	Grade	20.370638	20.0	-0.
25	25	initial_list_status_w	0.063553	2.545807e-06	initial	4.991815	5.0	0.
26	26	echeance_36	0.034634	1.338126e-02	echeance	2.720347	3.0	0.
27	27	ancianite_1a4	0.054371	3.200932e-04	ancianite	4.270596	4.0	-0.
28	28	ancianite_5a6	0.037888	9.647531e-02	ancianite	2.975965	3.0	0.
29	29	ancianite_7	0.025677	2.999673e-01	ancianite	2.016813	2.0	-0.
30	30	ancianite_8a9	0.002770	9.174683e-01	ancianite	0.217609	0.0	-0.
31	31	ancianite_10+	0.097862	1.549211e-11	ancianite	7.686666	8.0	0.
32	32	nb_mois_1er_credit_<87	1.813029	4.383749e-148	mois_credit	142.405533	142.0	-0.
33	33	nb_mois_1er_credit_87a89	1.486218	3.580371e-110	mois_credit	116.735933	117.0	0.
34	34	nb_mois_1er_credit_89a90	1.304625	4.417178e-76	mois_credit	102.472620	102.0	-0.
35	35	nb_mois_1er_credit_90a98	0.992666	2.705136e-55	mois_credit	77.969614	78.0	0.
36	36	nb_mois_1er_credit_98a101	0.693929	7.641018e-27	mois_credit	54.505140	55.0	0.
37	37	nb_mois_1er_credit_101a110	0.394082	4.202029e-10	mois_credit	30.953398	31.0	0.
38	38	nb_mois_1er_credit_110a126	0.072298	2.545465e-01	mois_credit	5.678658	6.0	0.

49	49	Revenu_126K-152K	0.548754	8.051498e-32	Revenu	43.102235	43.0	-0.
50	50	Revenu_152K-227K	0.564586	1.691447e-29	Revenu	44.345777	44.0	-0.
51	51	Revenu_>227K	0.602758	1.113899e-17	Revenu	47.343982	47.0	-0.
52	52	mths_since_last_delinq_null	0.305900	2.822434e-04	mths_since_last_delinq	24.027111	24.0	-0.
53	53	mths_since_last_delinq_0-4	0.289052	6.600199e-04	mths_since_last_delinq	22.703738	23.0	0.
54	54	mths_since_last_delinq_4-30	0.384151	6.219555e-06	mths_since_last_delinq	30.173383	30.0	-0.
55	55	mths_since_last_delinq_30-60	0.407221	1.783825e-06	mths_since_last_delinq	31.985398	32.0	0.
56	56	mths_since_last_delinq_60-83	0.350782	4.972709e-05	mths_since_last_delinq	27.552421	28.0	0.
57	57	total_acc_6-22	-0.063092	9.368739e-02	total_acc	-4.955568	-5.0	-0.
58	58	total_acc_22-50	-0.126080	1.237349e-03	total_acc	-9.903019	-10.0	-0.
59	59	total_acc_>50	-0.131218	1.009810e-02	total_acc	-10.306625	-10.0	0.
60	60	dti<=3.2	0.261988	2.977315e-11	dti	20.578017	21.0	0.
61	61	dti_3.2-8.8	0.301200	5.464292e-46	dti	23.657938	24.0	0.
62	62	dti_8.8-10.4	0.226738	7.638654e-17	dti	17.809319	18.0	0.
63	63	dti_10.4-13.6	0.187562	5.852196e-22	dti	14.732190	15.0	0.
64	64	dti_13.6-16.0	0.183184	2.298727e-19	dti	14.388279	14.0	-0.
65	65	dti_16.0-16.7	0.136840	2.773811e-05	dti	10.748175	11.0	0.
66	66	dti_16.7-19.9	0.105962	3.623664e-09	dti	8.322890	8.0	-0.
67	67	dti_19.9-20.8	0.094623	1.166047e-03	dti	7.432207	7.0	-0.
68	68	dti_20.8-23.2	0.038756	5.353195e-02	dti	3.044106	3.0	-0.
69	69	dti>35.2	-0.091007	1.921946e-01	dti	-7.148186	-7.0	0.
70	70	mths_since_last_record_null	0.188450	4.183036e-02	mths_since_last_record	14.801942	15.0	0.
71	71	mths_since_last_record_3-21	0.312915	1.638101e-02	mths_since_last_record	24.578132	25.0	0.
72	72	mths_since_last_record_21-31	0.305707	1.605904e-02	mths_since_last_record	24.011983	24.0	-0.
73	73	mths_since_last_record_31-85	0.379676	7.596651e-05	mths_since_last_record	29.821887	30.0	0.
74	74	mths_since_last_record>85	0.112414	2.366746e-01	mths_since_last_record	8.829646	9.0	0.
75	0	Home_RENT_ANY_OTHER_NONE	0.000000	NaN	Home	0.000000	0.0	0.
76	1	Adresse_ND_NE_IA_NV	0.000000	NaN	Adresse	0.000000	0.0	0.
77	2	Verification_Verified	0.000000	NaN	Verification	0.000000	0.0	0.
78	3	Purpose_ed_pyme_enerren_moving	0.000000	NaN	Purpose	0.000000	0.0	0.
79	4	Grades_F_G	0.000000	NaN	Grades	0.000000	0.0	0.
80	5	initial_list_status_f	0.000000	NaN	initial	0.000000	0.0	0.
81	6	echeance_60	0.000000	NaN	echeance	0.000000	0.0	0.
82	7	ancianite_<_1	0.000000	NaN	ancianite	0.000000	0.0	0.
83	8	nb_mois_1er_credit_>155	0.000000	NaN	mois_credit	0.000000	0.0	0.
84	9	Revenu_<20K	0.000000	NaN	Revenu	0.000000	0.0	0.
85	10	mths_since_last_delinq_83+	0.000000	NaN	mths_since_last_delinq	0.000000	0.0	0.
86	11	total_acc_<=6	0.000000	NaN	total_acc	0.000000	0.0	0.
87	12	dti_23.2-35.2	0.000000	NaN	dti	0.000000	0.0	0.
88	13	mths_since_last_record_0-3	0.000000	NaN	mths_since_last_record	0.000000	0.0	0.

```
In [225.. score_max_initial = df_scores.groupby('Variable Original')['Score Round'].max().sum()
```

```
In [226.. #Finalement le score maximum correspond.
score_max_initial
```

```
Out[226]: 850.0
```

```
In [227.. #De la même manière que nous avons estimé la probabilité de non-défaut (1-PD), pour estimer le score de crédit
#particulier, nous devons simplement additionner les scores pour lesquels ce client a des variables indicatrice
#Faisons un exercice pour illustrer :
```

```
In [228.. x_test.head()
```

Out[228]:

Unnamed: 0												
	id	member_id	loan_amnt	funded_amnt	funded_amnt_inv	term	int_rate	installment	grade	sub_grade		
89223	89223	7073644	8735123	9600	9600	9600.0	36 months	15.10	333.26	C	C2	J
158835	158835	3640390	4592970	14000	14000	14000.0	36 months	7.62	436.26	A	A3	Caref
108909	108909	6527485	8079529	16750	16750	16750.0	36 months	8.90	531.87	A	A5	Co
270155	270155	32419070	35032306	19750	19750	19750.0	60 months	20.20	525.46	E	E3	
23971	23971	606796	778433	15000	9475	9450.0	60 months	8.88	196.14	B	B1	E Inde

In [229]: df_scores

Out[229]:

	index	Variable Independante	Coefficientes	p_values	Variable Original	Calcul score	Score Round	Difference_in
0	0	Intercepto	-0.667983	NaN	Intercepto	322.542009	323.0	0.
1	1	Home_MORTGAGE	0.099254	4.826906e-15	Home	7.795976	8.0	0.
2	2	Home_OWEN	0.083285	3.966229e-05	Home	6.541680	7.0	0.
3	3	Adresse_FL	-0.207613	3.308666e-16	Adresse	-16.307117	-16.0	0.
4	4	Adresse_AL_HI_MO_NM	-0.156218	3.667267e-07	Adresse	-12.270275	-12.0	0.
5	5	Adresse_CA	-0.137447	1.484883e-10	Adresse	-10.795843	-11.0	-0.
6	6	Adresse_NC_ID_NJ	-0.140432	9.642841e-08	Adresse	-11.030333	-11.0	0.
7	7	Adresse_NY	-0.116864	2.028481e-06	Adresse	-9.179141	-9.0	0.
8	8	Adresse_KY_LA_MD	-0.123486	3.996527e-05	Adresse	-9.699258	-10.0	-0.
9	9	Adresse_MI_AR_AZ_VA_OK_DE_OH	-0.079535	3.239793e-04	Adresse	-6.247130	-6.0	0.
10	10	Adresse_MN_PA_UT_MA_RI_WA_TN_IN	-0.075609	5.771834e-04	Adresse	-5.938794	-6.0	-0.
11	11	Adresse_CT_IL	0.046172	1.162430e-01	Adresse	3.626584	4.0	0.
12	12	Adresse_TX	0.056687	3.165762e-02	Adresse	4.452541	4.0	-0.
13	13	Adresse_NH_AK_MT_MS_WY_WV_DC_ME	0.280750	1.870772e-10	Adresse	22.051728	22.0	-0.
14	14	Verification_Not Verified	0.095455	5.203965e-11	Verification	7.497551	7.0	-0.
15	15	Verification_Source Verified	-0.005208	7.020393e-01	Verification	-0.409071	-0.0	0.
16	16	Purpose_house_other_wedding_medical_vacation	0.389900	2.034026e-33	Purpose	30.624924	31.0	0.
17	17	Purpose_major_purchase_improvement_car	0.455846	5.272720e-42	Purpose	35.804741	36.0	0.
18	18	Purpose_debt_consolidation	0.400467	1.846630e-44	Purpose	31.454971	31.0	-0.
19	19	Purpose_credit_card	0.531613	1.113748e-66	Purpose	41.755855	42.0	0.
20	20	Grade_A	2.008301	0.000000e+00	Grade	157.743314	158.0	0.
21	21	Grade_B	1.323835	0.000000e+00	Grade	103.981454	104.0	0.
22	22	Grade_C	0.876908	8.340840e-291	Grade	68.877330	69.0	0.
23	23	Grade_D	0.553558	1.856190e-116	Grade	43.479607	43.0	-0.
24	24	Grade_E	0.259348	3.012767e-24	Grade	20.370638	20.0	-0.
25	25	initial_list_status_w	0.063553	2.545807e-06	initial	4.991815	5.0	0.
26	26	echeance_36	0.034634	1.338126e-02	echeance	2.720347	3.0	0.
27	27	ancianite_1a4	0.054371	3.200932e-04	ancianite	4.270596	4.0	-0.
28	28	ancianite_5a6	0.037888	9.647531e-02	ancianite	2.975965	3.0	0.
29	29	ancianite_7	0.025677	2.999673e-01	ancianite	2.016813	2.0	-0.
30	30	ancianite_8a9	0.002770	9.174683e-01	ancianite	0.217609	0.0	-0.
31	31	ancianite_10+	0.097862	1.549211e-11	ancianite	7.686666	8.0	0.

4.383749e-

86	11	total_acc_<=6	0.000000	NaN	total_acc	0.000000	0.0	0.0
87	12	dti_23.2-35.2	0.000000	NaN	dti	0.000000	0.0	0.0
88	13	mths_since_last_record_0-3	0.000000	NaN	mths_since_last_record	0.000000	0.0	0.0

```
In [230...] #Score de crédit pour une observation (même exercice que la ligne 170)
322+7+42+69+5+4+7+3+4+55+2+28+5+15
```

```
Out[230]: 558
```

```
In [231...] #Maintenant, calculons le score de crédit pour tous les clients de la base d'évaluation.
#Nous devons multiplier la matrice d'indicateurs (0-1) par les scores qui doivent avoir la même dimension.
```

```
In [232...] x_test_model.head()
```

```
Out[232]:
```

	Home_MORTGAGE	Home_RENT_ANY_OTHER_NONE	Home_OWN	Adresse_ND_NE_IA_NV	Adresse_FL	Adresse_AL_HI_MO_NM	Adr
89223	0	0	1	0	0	0	
158835	1	0	0	0	0	1	
108909	1	0	0	0	0	0	
270155	0	1	0	0	0	0	
23971	0	1	0	0	0	0	

```
In [233...] #insérer une colonne pour l'intercept pour pouvoir multiplier
```

```
In [234...] x_test_intercepto = x_test_model
x_test_model.insert(0, 'Intercepto', 1)
```

```
In [235...] x_test_intercepto.head()
```

```
Out[235]:
```

	Intercepto	Home_MORTGAGE	Home_RENT_ANY_OTHER_NONE	Home_OWN	Adresse_ND_NE_IA_NV	Adresse_FL	Adresse_AL_HI_M
89223	1	0	0	1	0	0	
158835	1	1	0	0	0	0	
108909	1	1	0	0	0	0	
270155	1	0	1	0	0	0	
23971	1	0	1	0	0	0	

```
In [236...] #Para garantizar que les indicateurs et les scores soient dans le même ordre avant de les multiplier, fixons l'
#x_test_con_intercepto égal à l'ordre de la colonne 'Variable Indépendante' de df_scores.
```

```
In [237...] #Prends le même ordre que la colonne "Variable Indépendante" du DataFrame df_scores.
x_test_intercepto = x_test_intercepto[df_scores['Variable Indépendante'].values]
```

```
In [238...] x_test_intercepto.head()
```

```
Out[238]:
```

	Intercepto	Home_MORTGAGE	Home_OWN	Adresse_FL	Adresse_AL_HI_MO_NM	Adresse_CA	Adresse_NC_ID_NJ	Adresse_NY	Ad
89223	1	0	1	0	0	0	0	0	
158835	1	1	0	0	1	0	0	0	
108909	1	1	0	0	0	0	0	0	
270155	1	0	0	0	0	1	0	0	
23971	1	0	0	0	0	1	0	0	

```
In [239...] #les scores dans une nouvelle variable
```

```
In [240...] scores_round = df_scores['Score Round']
```

```
In [241...] #S'Assurer qu'ils ont la même dimension.
```

```
In [242...] x_test_intercepto.shape
```

```
Out[242]: (46629, 89)
```

```
In [243...] scores_round.shape
```

```
Out[243]: (89,)
```

```
In [244... #Pour les rendre compatibles en leur dimension
```

```
In [245... scores_round = scores_round.values.reshape(89,1)
```

```
In [246... scores_round.shape
```

```
Out[246]: (89, 1)
```

```
In [247... #multiplier toutes les lignes de deux matrices,puis à additionner toutes ces multiplications.
```

```
In [248... y_scores = x_test_intercepto.dot(scores_round)
```

```
In [249... y_scores.head()
```

```
Out[249]:
```

	0
89223	559.0
158835	631.0
108909	588.0
270155	537.0
23971	514.0

```
In [250... # Nous avons obtenu presque le même score que celui que nous avons estimé "manuellement" pour l'observation 892.
```

De Puntajes Crediticios a Probabilidad de Default

```
In [251... #Estimer la somme des coefficients à partir des scores de crédit.
#Coefficients = ((Calcul du score - score_min)/(score_max - score_min)) * (somme_coefficient_max - somme_coeffi
```

```
In [252... somme_coef_from_scores = ((y_scores - score_min)/(score_max - score_min)) * (somme_coeficiente_max - somme_coef
```

```
In [253... #Calcul dela probabilité.

#Nous avons sommé les coefficients dans le modèle PD, nous avons obtenu le logarithme des cotes :
#ln((1-PD)/PD) = 2.308207. Ensuite, nous avons appliqué l'exponentielle des deux côtés pour obtenir (1-PD)/PD e
#pour PD :1 - PD = 10.056377395809957 / (10.056377395809957 + 1) Pour obtenir 1 - PD, alors, ici on doit faire
#Au lieu d'appliquer le logarithme, nous devons appliquer l'exponentielle.
```

```
In [254... y_proba_from_scores = np.exp(somme_coef_from_scores) / (np.exp(somme_coef_from_scores) + 1)
```

```
In [255... #proba no default
y_proba_from_scores.head()
```

```
Out[255]:
```

	0
89223	0.912334
158835	0.963000
108909	0.937712
270155	0.887192
23971	0.854402

```
In [256... #ces probabilités sont similaires à celles que nous avons obtenues directement à partir du modèle PD, qui sont
#proches mais changent en raison de l'arrondi que nous avons effectué précédemment.
```

```
In [257... proba_bon[0:5]
```

```
Out[257]: array([0.90955448, 0.96180255, 0.9367632 , 0.88682306, 0.85416882])
```

Établir le Seuil critique

```
In [258... #Lorsque nous avons généré la courbe ROC, nous avons créé un tableau avec les seuil
```

Seuil critique pour définir à qui prêter et à qui ne pas prêter

```
In [259... seuil
```

```
Out[259]: array([          inf, 0.99566605, 0.99349452, ..., 0.53623792, 0.48965534,
          0.470524  ])
```

```
In [260... seuil.shape
```

```
Out[260]: (7726,)
```

```
In [261... # Nous créons un dataframe où nous sauvegarderons les données pour sélectionner les seuil critique.
```

```
In [262... df_seuil = pd.concat([pd.DataFrame(seuil), pd.DataFrame(faux_positifs), pd.DataFrame(vrai_positifs)], axis = 1)
```

```
In [263... df_seuil.head()
```

```
Out[263]:
```

	0	0	0
0	inf	0.000000	0.000000
1	0.995666	0.000000	0.000024
2	0.993495	0.000000	0.001681
3	0.993490	0.000228	0.001681
4	0.992466	0.000228	0.003551

```
In [264... df_seuil.columns = ['seuil', 'faux_positifs', 'vrai_positifs']
```

```
In [265... df_seuil.head()
```

```
Out[265]:
```

	seuil	faux_positifs	vrai_positifs
0	inf	0.000000	0.000000
1	0.995666	0.000000	0.000024
2	0.993495	0.000000	0.001681
3	0.993490	0.000228	0.001681
4	0.992466	0.000228	0.003551

```
In [266... # Le premier point de coupure n'a pas de sens, il est estimé par défaut comme infini ou comme (1 + le deuxième
```

```
In [267... df_seuil.loc[0, 'seuil'] = 1
```

```
In [268... df_seuil.head()
```

```
Out[268]:
```

	seuil	faux_positifs	vrai_positifs
0	1.000000	0.000000	0.000000
1	0.995666	0.000000	0.000024
2	0.993495	0.000000	0.001681
3	0.993490	0.000228	0.001681
4	0.992466	0.000228	0.003551

Estimant le seuil comme un score de crédit à partir de la proba de no défaut (1-P)

```
In [270... # Credit Score = (ln(1-P/P) - somme_coeff_min) * (score_max - score_min) / (somme_coeff_max - somme_coeff_min)
```

```
In [271... df_seuil['Credit score'] = ((np.log(df_seuil['seuil'] / (1 - df_seuil['seuil'])) - somme_coeficiente_min) * ((s
```

```
In [272... df_seuil.head()
```

```
Out[272]:
```

	seuil	faux_positifs	vrai_positifs	Credit score
0	1.000000	0.000000	0.000000	inf
1	0.995666	0.000000	0.000024	802.0
2	0.993495	0.000000	0.001681	770.0
3	0.993490	0.000228	0.001681	770.0
4	0.992466	0.000228	0.003551	758.0

```
In [273... # Définissons le score pour le point de coupure de 1.00 comme le maximum (850)
```

```
In [274... df_seuil.loc[0, 'Credit score'] = score_max
```

```
In [275... df_seuil.head()
```

Out[275]:

	seuil	faux_positifs	vrai_positifs	Credit score
0	1.000000	0.000000	0.000000	850.0
1	0.995666	0.000000	0.000024	802.0
2	0.993495	0.000000	0.001681	770.0
3	0.993490	0.000228	0.001681	770.0
4	0.992466	0.000228	0.003551	758.0

In [276...

#dernières observations

In [277...

df_seuil.tail()

Out[277]:

	seuil	faux_positifs	vrai_positifs	Credit score
7721	0.551097	0.999315	0.999858	391.0
7722	0.541944	0.999315	0.999905	388.0
7723	0.536238	0.999772	0.999905	386.0
7724	0.489655	0.999772	1.000000	372.0
7725	0.470524	1.000000	1.000000	366.0

In [278...

#Maintenant, regardons les taux d'approbation et de rejet pour chaque seuil. Pour estimer ces taux, nous avons

#Le nombre de demandes approuvées
#Le nombre de demandes rejetées
#Le nombre total de demandes
#Total des demandes = Nombre d'approuvées + Nombre de rejetées
#Taux d'approbation = Nombre d'approuvées / Total des demandes
#Taux de rejet = 1 - Taux d'approbation

In [279...

#fonction qui calcule le nombre de candidats approuvés pour un seuil donné.
def n_approuvés(p): # où p c'est la proba du seuil
 return np.where(df_predict['Proba_Preditcion'] >= p, 1, 0).sum()

In [280...

Nb d'approuvés

In [281...

df_seuil['approuvés'] = df_seuil['seuil'].apply(n_approuvés)

In [282...

Nb Non_approuvés: Total de Aplicaciones - Nb d'approuvés

In [283...

df_seuil['Non_approuvés'] = df_predict['Proba_Preditcion'].shape[0] - df_seuil['approuvés']

In [284...

Taux approuvés = Nb approuvés / Total de Aplicaciones

In [285...

df_seuil['Taux approuvés'] = df_seuil['approuvés'] / df_predict['Proba_Preditcion'].shape[0]

In [286...

Taux Non approuvés = 1 - Tasa de Aprobación

In [287...

df_seuil['Taux non approuvés'] = 1 - df_seuil['Taux approuvés']

In [288...

df_seuil.head()

Out[288]:

	seuil	faux_positifs	vrai_positifs	Credit score	approuvés	Non_approuvés	Taux approuvés	Taux non approuvés
0	1.000000	0.000000	0.000000	850.0	0	46629	0.000000	1.000000
1	0.995666	0.000000	0.000024	802.0	1	46628	0.000021	0.999979
2	0.993495	0.000000	0.001681	770.0	71	46558	0.001523	0.998477
3	0.993490	0.000228	0.001681	770.0	72	46557	0.001544	0.998456
4	0.992466	0.000228	0.003551	758.0	151	46478	0.003238	0.996762

Interpretation des résultats

¿Quel serait le score de crédit et le taux d'approbation pour une probabilité de non-défaut de 95%?

Réponse à la ligne 864 se trouve un 1-PD de 95,0037% avec un score de 606 et un taux d'approbation de 29,8%.

In [290...

df_seuil

Out[290]:

	seuil	faux_positifs	vrai_positifs	Credit score	approuvés	Non_approuvés	Taux approuvés	Taux non approuvés
0	1.000000	0.000000	0.000000	850.0	0	46629	0.000000	1.000000
1	0.995666	0.000000	0.000024	802.0	1	46628	0.000021	0.999979
2	0.993495	0.000000	0.001681	770.0	71	46558	0.001523	0.998477
3	0.993490	0.000228	0.001681	770.0	72	46557	0.001544	0.998456
4	0.992466	0.000228	0.003551	758.0	151	46478	0.003238	0.996762

7722	0.541944	0.999315	0.999905	388.0	46622	7	0.999850	0.000150
7723	0.536238	0.999772	0.999905	386.0	46624	5	0.999893	0.000107
7724	0.489655	0.999772	1.000000	372.0	46628	1	0.999979	0.000021
7725	0.470524	1.000000	1.000000	366.0	46629	0	1.000000	0.000000

In [291... `# Exportemos tablas a csv`

In [291... `#x_test_intercepto.to_csv('x_test_intercepto.csv')`

In [292... `#df_seuil.to_csv('df_seuil.csv')`

In [293... `#df_scores.to_csv('df_scores.csv')`