المندوبية السامية للتخطيط
HAUT-COMMISSARIAT AU PLAN

INSEA

Institut National de Statistiques et d'Economie Appliqué

Segmentation des clients de La banque

Réalisé par :

Meryem Belkhayat

Khadija Ben Talha

Encadré par :

Kaoutar ELHARI

Année universitaire 2021/2022

Sommaire

I. Introduction :

Pour faire du profit. Le succès de la banque est directement lié à sa capacité à contrôler et à gérer les risques associés. Les banques sont exposées à différents types de risques, mais le risque le plus difficile qui peut entraîner une faillite complète d'une banque est le risque de crédit. Le risque de crédit est un sujet important et largement étudié dans les décisions de prêt et la rentabilité du secteur bancaire. Pour toutes les banques, le crédit reste le principal risque difficile à compenser.

Habituellement, l'approche générique de l'évaluation du risque de crédit consiste à appliquer des algorithmes de classification sur des données similaires de clients précédents, clients fidèles et irresponsables, afin de trouver une relation entre la caractéristique et les défaillances potentielles. Afin de classer les candidats en deux groupes : les candidats ayant un bon crédit et les candidats ayant un mauvais crédit. Les candidats ayant un bon crédit ont une grande possibilité de rembourser leurs obligations financières tandis que les candidats ayant un mauvais crédit ont une forte possibilité de défaut.

II. Présentation du projet :

Ce projet est réalisé dans un contexte pédagogique sous le sujet de segmentation des clients de la banque, il présente une analyse des données de crédit allemandes qui contiennent les détails de 1000 demandeurs de prêt avec 10 attributs et la classification si un client est un bon payant ou non.

Dans ce projet, la relation entre le risque de crédit et divers attributs sera explorée à l'aide de techniques statistiques de base et présentée à l'aide de visualisations…

III. Base de données :

L'ensemble de données allemandes contient 1000 entrées avec 20 attributs catégoriels/symboliques préparés par le professeur Hofmann. Dans cet ensemble de données, chaque entrée représente une personne qui prend un crédit auprès d'une banque. Chaque personne est classée comme bon ou mauvais risque de crédit selon l'ensemble d'attributs.

	Age	Sex	Job	Housing	Saving accounts	Checking account	Credit amount	Duration	Purpose	Risk
0	67	male	2	own	NaN	little	1169	6	radio/TV	good
1	22	female	2	own	little	moderate	5951	48	radio/TV	bad
2	49	male	1	own	little	NaN	2096	12	education	good
3	45	male	2	free	little	little	7882	42	furniture/equipment	good
4	53	male	2	free	little	little	4870	24	car	bad
...	...	...	...	...	...	...	...	...	...	...
995	31	female	1	own	little	NaN	1736	12	furniture/equipment	good
996	40	male	3	own	little	little	3857	30	car	good
997	38	male	2	own	little	NaN	804	12	radio/TV	good
998	23	male	2	free	little	little	1845	45	radio/TV	bad
999	27	male	2	own	moderate	moderate	4576	45	car	good

1000 rows × 10 columns

- **Age :** variable numérique, qui représente l'age du client
- **Sex :** variable catégorielle, qui prend les valeurs : female/male
- **Job :** variable numérique, qui prend 4 valeurs :
    - 0 :unskilled and non-resident
    - 1 :unskilled and resident
    - 2 :skilled
    - 3 :highly skilled
- **Housing :** variable catégorielle, elle prend la valeur own, rent ou free
- **Saving accounts :** variable catégorielle, qui prend 4 valeurs little, moderate, quite rich or rich
    - Mark little : ... < 100 DM
    - moderate : 100 <= ... < 500 DM
    - quite rich : 500 <= ... < 1000 DM
    - rich : .. >= 1000 DM
    - None : unknown/ no savings account
- **Checking account :** variable catégorielle, qui prend 4 valeurs little, moderate or rich
    - little : ... < 0 DM
    - moderate : 0 <= ... < 200 DM
    - rich : ... >= 200 DM / salary assignments for at least 1 year
    - None : unknown/ no checking account
- **Credit amount :** variable numérique en DM (Deutsch Mark)
- **Duration :** variable numérique en mois
- **Purpose :** variable catégorielle, elle représente le but du crédit (car, furniture/ equipment, radio/TV, domestic appliances, repairs, education, business,vacation/ others)
- **Risk :** colonne cible, qui classifié les candidats ayant un bon crédit et les candidats ayant un mauvais crédit (good/ bad)

IV. L'analyse exploratoire des données (AED) :

- ### Data checks :

Deux colonnes contiennent des valeurs manquantes :

```
[ ]  credit.info()

     <class 'pandas.core.frame.DataFrame'>
     RangeIndex: 1000 entries, 0 to 999
     Data columns (total 10 columns):
      #   Column            Non-Null Count  Dtype
     ---  ------            --------------  -----
      0   Age               1000 non-null   int64
      1   Sex               1000 non-null   object
      2   Job               1000 non-null   int64
      3   Housing           1000 non-null   object
      4   Saving accounts   817 non-null    object
      5   Checking account  606 non-null    object
      6   Credit amount     1000 non-null   int64
      7   Duration          1000 non-null   int64
      8   Purpose           1000 non-null   object
      9   Risk              1000 non-null   object
     dtypes: int64(4), object(6)
     memory usage: 78.2+ KB
```

Les données statistiques des variables numériques :

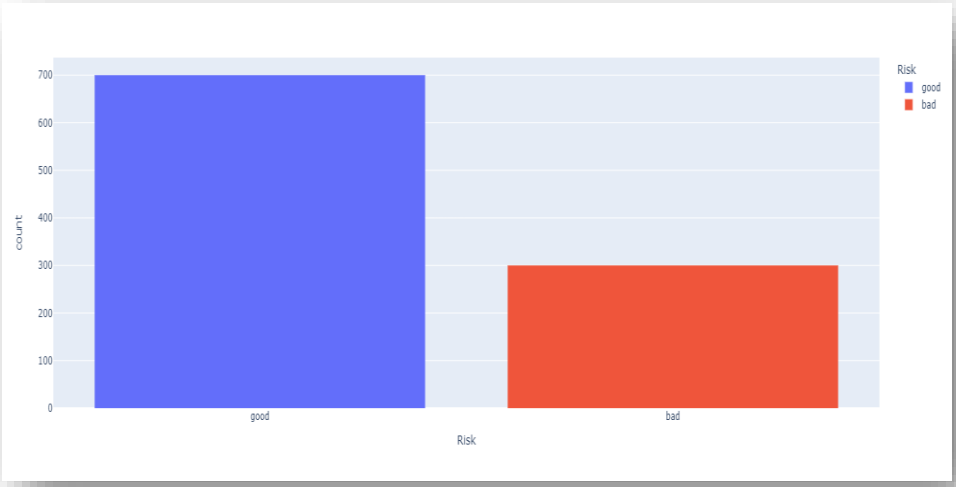
```
credit.describe()
```

	Age	Job	Credit amount	Duration
count	1000.000000	1000.000000	1000.000000	1000.000000
mean	35.546000	1.904000	3271.258000	20.903000
std	11.375469	0.653614	2822.736876	12.058814
min	19.000000	0.000000	250.000000	4.000000
25%	27.000000	2.000000	1365.500000	12.000000
50%	33.000000	2.000000	2319.500000	18.000000
75%	42.000000	2.000000	3972.250000	24.000000
max	75.000000	3.000000	18424.000000	72.000000

- **Analyse descriptive :**

L'analyse exploratoire des données est un moyen nécessaire et puissant d'explorer un jeu de données. Elle est utilisée pour analyser et étudier les ensembles de données puis résumer leurs principales caractéristiques, souvent en employant des méthodes de visualisation des données. Ainsi, on peut découvrir plus facilement des modèles (patterns), identifier des anomalies, tester une hypothèse ou vérifier des suppositions.

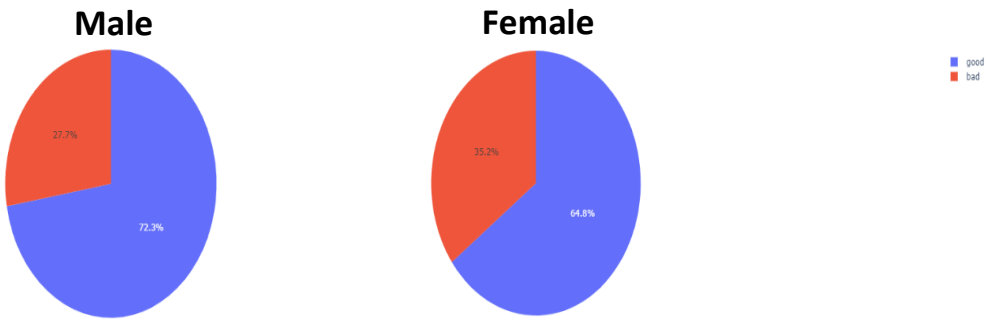
➢ **Distribution de la variable cible :**



Le diagramme montre que la répartition des clients ayant un bon crédit et ceux ayant un mauvais crédit n'est pas équilibré. Il s'agit donc d'un problème de classe déséquilibrée. Il faut opter pour un rééchantillonnage (Bootstrap) afin de créer des « nouveaux échantillons » statistiques à partir de l'échantillon initial.

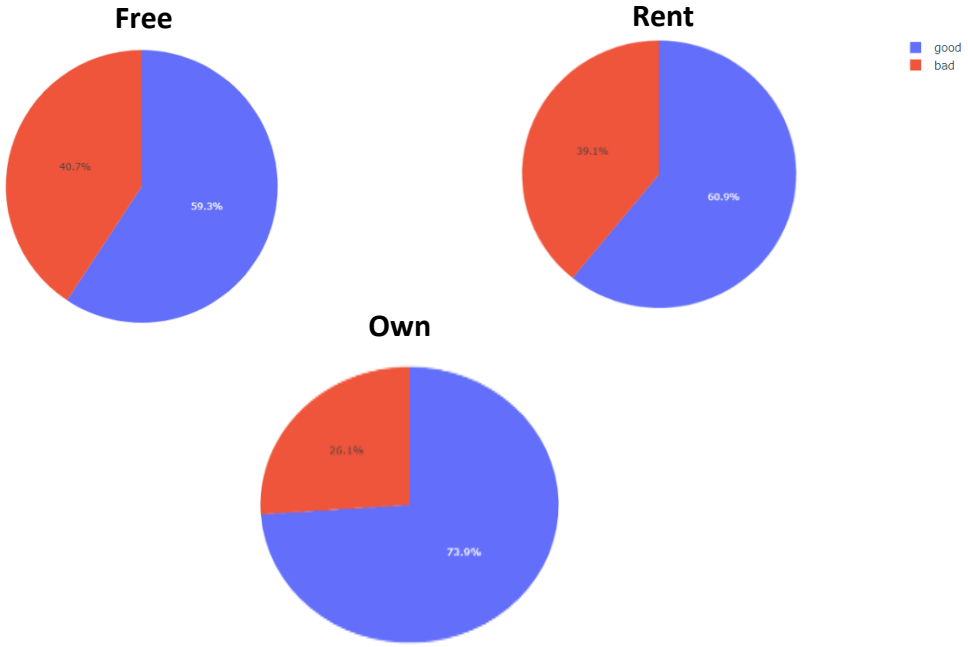
Ce processus sera appliqué par la suite.

- **Relation sex et risk :**



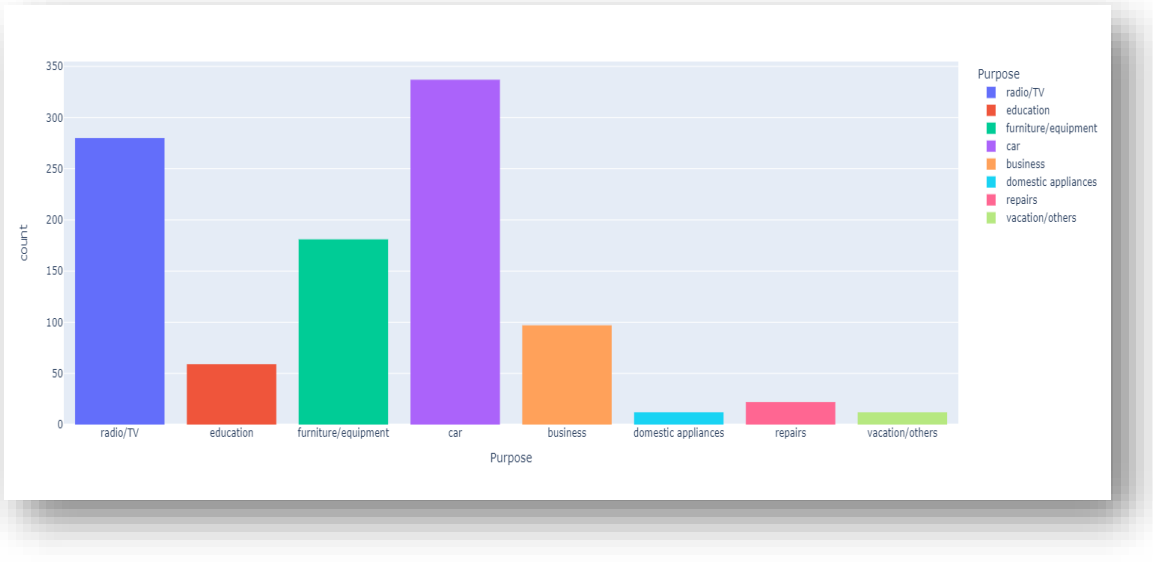
Les clients homme sont des bons payeurs de crédit de 7.5% plus que les clients femmes, ce qui renforce la contribution de la variable Sex dans la classification des clients.

- **Relation housing et risk :**

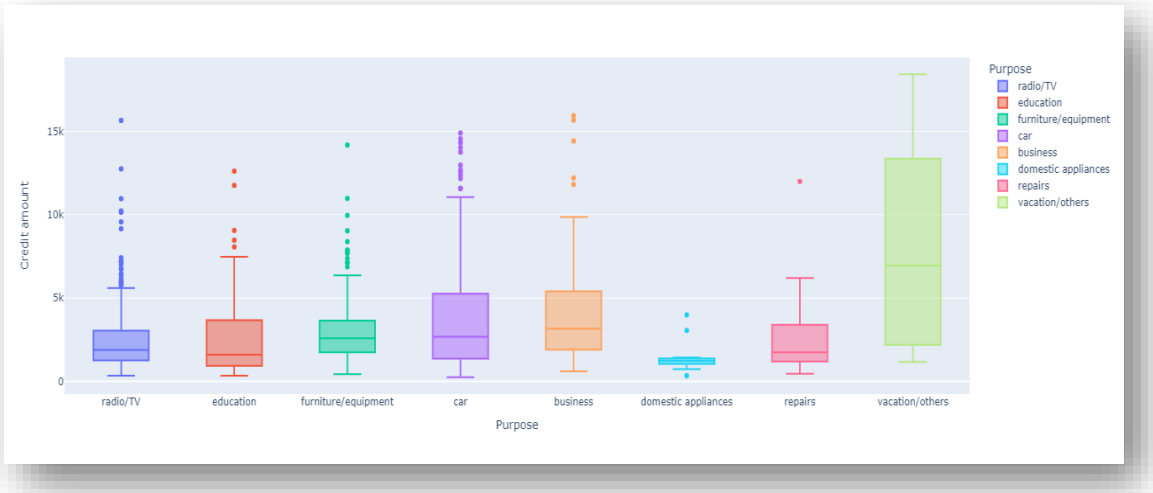


La répartition des clients ayant un bon crédit et ceux ayant un mauvais crédit se diffère d'un type d'habitat à un autre. Donc la variable 'Housing' influence la variable cible.

➢ **Type de crédit :**

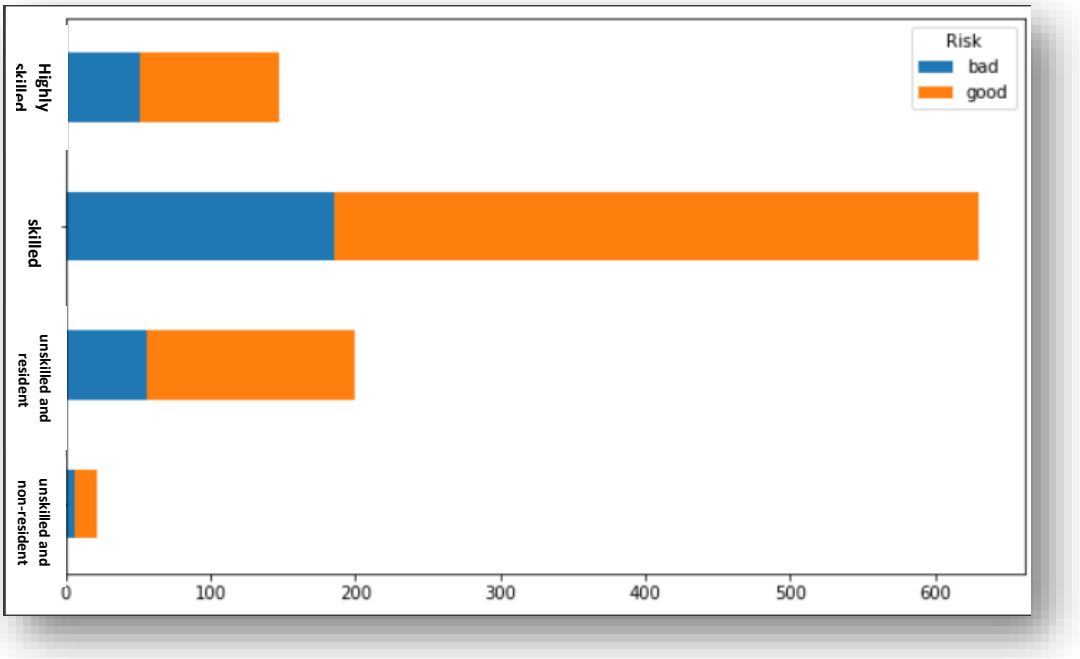


L'histogramme montre que les types de crédit les plus prélevés sont liés à l'achat d'une voiture, suivi par la radio/TV.



Le graphique ci-dessus montre que les montants du crédit les plus élevés sont prélevés pour les vacances/autres, les plus petits pour les appareils électroménagers. La plupart d'entre eux ont des valeurs aberrantes sur le côté supérieur des cases, qui seront éliminer par la suite.

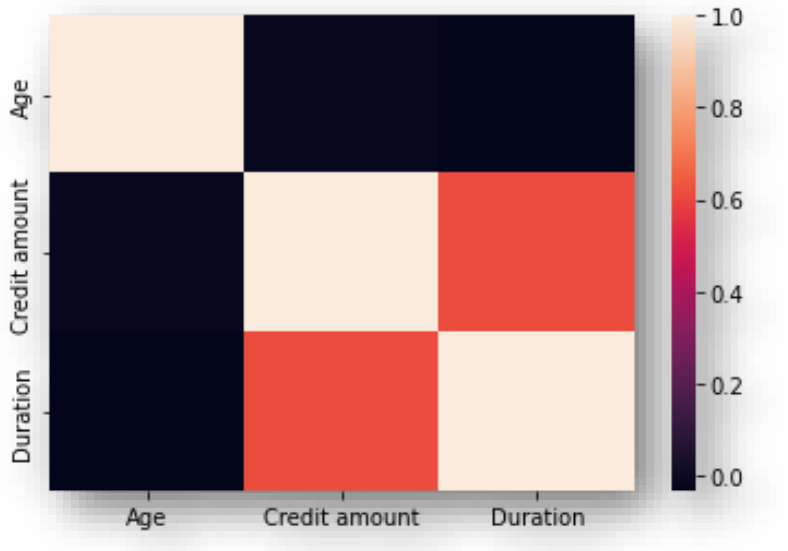
➢ Relation Job et Risk:



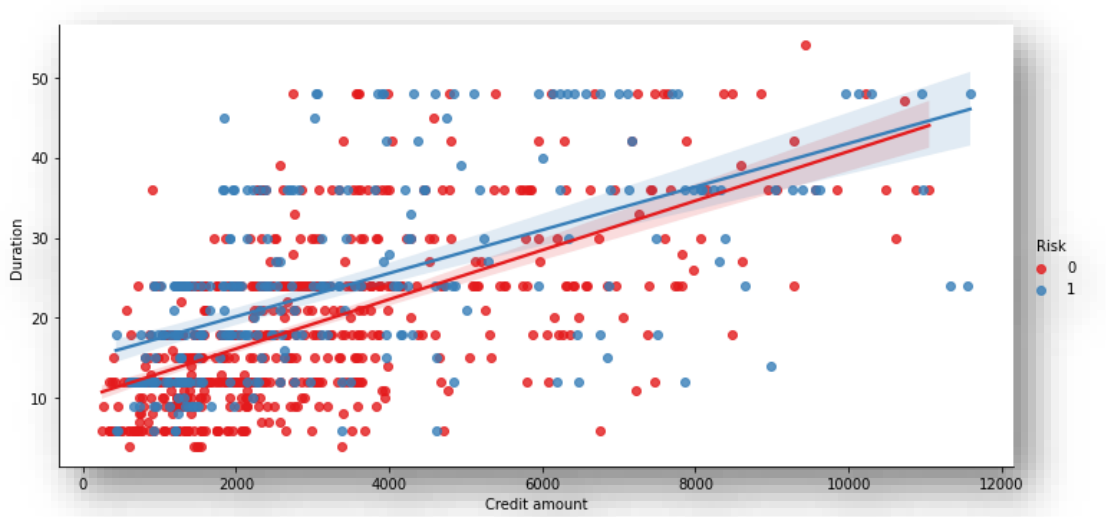
le graphique montre que les candidats umeployed/unskilled (chômage/non qualifiés) présentent un risque élevé.

➢ Corrélation des variables numériques :

Les variables Duration et Credit amount semblent être corrélées entre eux :



Explorant d'avantage cette corrélation :



Il existe une relation linéaire entre le montant du crédit et sa durée. Le graphe montre qu'en général, les crédits les plus élevés ont une durée de remboursement plus longue. Les cas où des crédits importants sont accordés avec une courte période de remboursement se sont avérés être des prêts irrécouvrables.

Cette corrélation pose un problème de colinéarité des variables explicatives, ce qui conduit à une mauvaise classification. On choisit, par la suite, celle qui contribue le plus dans la construction du bon modèle.

V. L'ingénierie des fonctionnalités (Feature engineering) :

L'ingénierie des fonctionnalités est un processus permettant de sélectionner et de transformer des variables lors de la création d'un modèle prédictif à l'aide du machine Learning ou de la modélisation statistique. L'ingénierie des fonctions inclut généralement la création, la transformation, l'extraction et la sélection de fonctionnalités.

➢ Valeurs manquantes :

Les valeurs manquantes, identifié précédament dans la phase de Data checks, signifient que le client n'a pas de checking account/savings account ou bien sa situation est inconnue. Pour cela on a introduit une nouvelle catégorie 'Others' :

```
credit['Saving accounts'] = credit['Saving accounts'].fillna('Others')
credit['Checking account'] = credit['Checking account'].fillna('Others')
```

➢ La transformation de fonctionnalités :

Un modèle d'apprentissage automatique ne peut malheureusement pas traiter les variables catégorielles. Par conséquent, on doit trouver un moyen d'encoder ces variables sous forme numérique avant de les employer dans le modèle.

Il existe deux manières principales de mener à bien ce processus :

- Label Encoding: est le concept d'attribution de chaque catégorie unique dans une variable catégorielle avec un entier. Aucune nouvelle colonne n'est créée.

Le risque à prédire : soit 0 pour le crédit présentant aucun risque et sera remboursé dans les délais, soit 1 indiquant que le crédit présente un risque et que le client rencontrera des difficultés de paiement.

```
credit['Risk'] = credit['Risk'].replace(["good"],0)
credit['Risk'] = credit['Risk'].replace(["bad"],1)
```

```
credit
```

	Age	Sex	Job	Housing	Saving accounts	Checking account	Credit amount	Duration	Purpose	Risk
0	67	male	2	own	Others	little	1169	6	radio/TV	0
1	22	female	2	own	little	moderate	5951	48	radio/TV	1
2	49	male	1	own	little	Others	2096	12	education	0
3	45	male	2	free	little	little	7882	42	furniture/equipment	0
4	53	male	2	free	little	little	4870	24	car	1
...	...	...	...	...	...	...	...	...	...	...
995	31	female	1	own	little	Others	1736	12	furniture/equipment	0
996	40	male	3	own	little	little	3857	30	car	0
997	38	male	2	own	little	Others	804	12	radio/TV	0
998	23	male	2	free	little	little	1845	45	radio/TV	1
999	27	male	2	own	moderate	moderate	4576	45	car	0

958 rows × 10 columns

- One-Hot Encoding : est le concept de création d'une nouvelle colonne pour chaque catégorie unique dans une variable catégorique. Chaque observation reçoit la valeur «1» dans la colonne de sa catégorie correspondante et la valeur « 0 » dans toutes les autres nouvelles colonnes.

```
cat_features = ['Sex', 'Job','Housing', 'Saving accounts', 'Checking account','Purpose']
num_features=['Age', 'Credit amount', 'Duration','Risk']
for variable in cat_features:
    dummies = pd.get_dummies(credit[cat_features])
    df1= pd.concat([credit[num_features], dummies],axis=1)

df1
```

	Age	Credit amount	Duration	Risk	Job	Sex_female	Sex_male	Housing_free	Housing_own	Housing_rent	...	Checking account_moderate	Checking account_rich	Purpose_business	Purpose_car	Purpose_domestic appliances	Purpose_education	P
0	67	1169	6	0	2	0	1	0	1	0	...	0	0	0	0	0	0	
1	22	5951	48	1	2	1	0	0	1	0	...	1	0	0	0	0	0	
2	49	2096	12	0	1	0	1	0	1	0	...	0	0	0	0	0	1	
3	45	7882	42	0	2	0	1	1	0	0	...	0	0	0	0	0	0	
4	53	4870	24	1	2	0	1	1	0	0	...	0	0	0	1	0	0	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
995	31	1736	12	0	1	1	0	0	1	0	...	0	0	0	0	0	0	
996	40	3857	30	0	3	0	1	0	1	0	...	0	0	0	1	0	0	
997	38	804	12	0	2	0	1	0	1	0	...	0	0	0	0	0	0	
998	23	1845	45	1	2	0	1	1	0	0	...	0	0	0	0	0	0	
999	27	4576	45	0	2	0	1	0	1	0	...	1	0	0	1	0	0	

1000 rows × 27 columns

➢ Valeurs aberrantes :

Le score Z, ou score standard, est un moyen de décrire un point de données selon sa relation avec la moyenne et l'écart-type d'un groupe de points. L'établissement du score

Z revient simplement à représenter les données dans une distribution dont la moyenne est définie sur 0 et dont l'écart-type est défini sur 1.

L'objectif du score Z est d'éliminer les effets de localisation et d'échelle des données, ce qui permet de comparer directement des ensembles de données différents. L'idée derrière la méthode de score Z pour détecter les valeurs aberrantes est qu'une fois que vous avez centré et remis à l'échelle les données, toute valeur trop éloignée de zéro (le seuil est généralement un score Z de 3 ou -3) est considérée comme aberrante.

```
credit_copy=df1
credit_copy=credit_copy[num_features].drop(['Risk'],axis=1)
z = np.abs(stats.zscore(credit_copy))
df1 = df1[(z < 3).all(axis=1)]
df1
```

	Age	Credit amount	Duration	Risk	Job	Sex_female	Sex_male	Housing_free	Housing_own	Housing_rent	...	Checking account_moderate	Checking account_rich	Purpose_business	Purpose_car	Purpose_domestic appliances	Purpose_education	P
0	67	1169	6	0	2	0	1	0	1	0	...	0	0	0	0	0	0	
1	22	5951	48	1	2	1	0	0	1	0	...	1	0	0	0	0	0	
2	49	2096	12	0	1	0	1	0	1	0	...	0	0	0	0	0	1	
3	45	7882	42	0	2	0	1	1	0	0	...	0	0	0	0	0	0	
4	53	4870	24	1	2	0	1	1	0	0	...	0	0	0	1	0	0	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
995	31	1736	12	0	1	1	0	0	1	0	...	0	0	0	0	0	0	
996	40	3857	30	0	3	0	1	0	1	0	...	0	0	0	1	0	0	
997	38	804	12	0	2	0	1	0	1	0	...	0	0	0	0	0	0	
998	23	1845	45	1	2	0	1	1	0	0	...	0	0	0	0	0	0	
999	27	4576	45	0	2	0	1	0	1	0	...	1	0	0	1	0	0	

958 rows × 27 columns

Après l'application de cette approche, on remarque la suppression de 42 valeurs aberrantes.

➢ Rééchantillonnage des données :

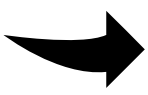
Le SMOTE, acronyme pour Synthetic Minority Oversampling Technique, est une méthode de suréchantillonnage des observations minoritaires. Pour éviter de réaliser un simple clonage des individus minoritaires, le SMOTE se base sur un principe simple : générer de nouveaux individus minoritaires qui ressemblent aux autres, sans être strictement identiques. Cela permet de densifier de façon plus homogène la population d'individus minoritaires.

Alors, on élimine la colonne cible 'Risk' de dataframe, afin de procéder au rééchantillonnage des données, par la fonction Smote :

```
y= df1['Risk']
x= df1.drop(['Risk','Credit amount'],axis=1)
from imblearn.over_sampling import SMOTE
oversample = SMOTE()
x, y= oversample.fit_resample(x, y)
```

On peut remarquer l'augmentation de nombre d'observations, après et avant le rééchantillonnage :

```
x.shape, y.shape
((1362, 25), (1362,))
```



```
x.shape, y.shape
((958, 25), (958,))
```

VI. Modèle d'entrainement, test et évaluation :

1) Répartir la base des données en ensembles d'apprentissage et de test :

À l'aide de la fonction train_test_split, on répartit notre base de données en ensemble d'apprentissage (80%) et un de test (20%).

```
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.20,random_state = 30)

[ ] x_train.shape, x_test.shape

    ((1089, 25), (273, 25))
```

2) La mise à échelle de la data

La mise à l'échelle des fonctionnalités consiste à transformer les valeurs de différentes fonctionnalités numériques pour qu'elles se situent dans une plage similaire les unes aux autres. La mise à échelle des fonctionnalités est utilisée pour empêcher les modèles d'apprentissage supervisé d'être biaisés vers une plage de valeurs spécifique.

Dans notre cas, on va utiliser la fonction MinMaxScaler:

3) Sélection de modèle:

Avant de sélectionner le bon algorithme pour notre classification, on élimine la variable 'Credit amount', due sa forte corrélation avec la variable 'Duration' et puisqu'elle impacte moins la variable cible 'Risk' :

```
y= df1['Risk']
x=df1.drop(['Risk','Credit amount'],axis=1)
```

- **Répartir la data d'entrainement en ensembles d'apprentissage et de validation:**

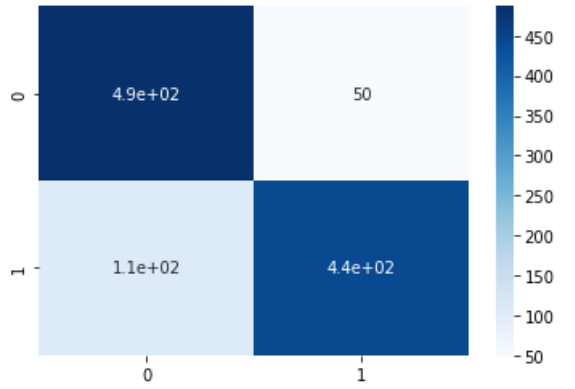
```
[39] x_train1,x_valid,y_train1,y_valid = train_test_split(x_train,y_train,test_size=0.20,random_state = 31)
```

- **Entrainement des modèles condidats:**

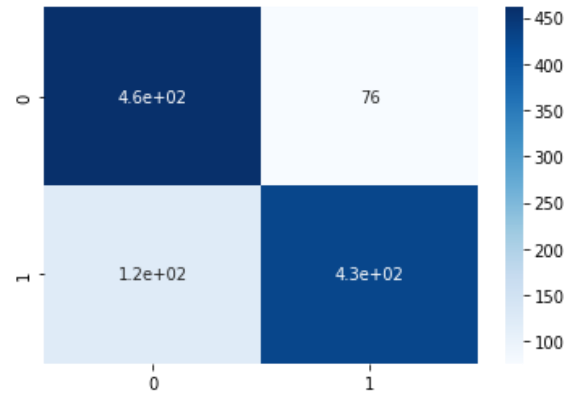
```
[40] models=[SVC(),LogisticRegression(),LinearDiscriminantAnalysis(),RandomForestClassifier()]

    for model in models:
        model.fit(x_train1,y_train1)
        y_pred=model.predict(x_train1)
        sns.heatmap(confusion_matrix(y_train1,y_pred),annot=True,cmap=plt.cm.Blues)
        plt.show()
        print(classification_report(y_train1,y_pred))
```

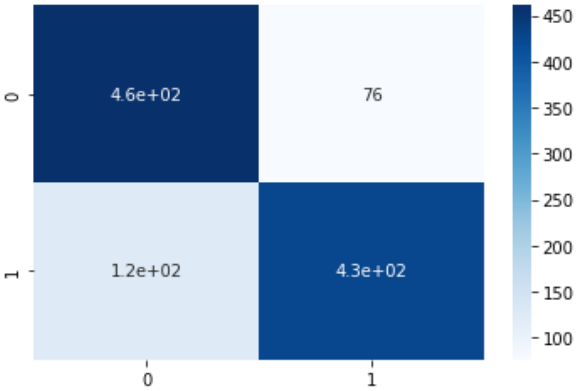
Les résultats de matrice de confusion et de précision pr chaque modèlesont,respectivement, les suivantes :



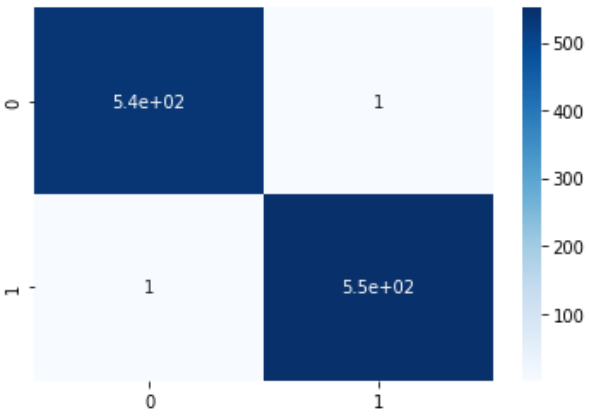
	precision	recall	f1-score	support
0	0.82	0.91	0.86	537
1	0.90	0.81	0.85	552
accuracy			0.86	1089
macro avg	0.86	0.86	0.86	1089
weighted avg	0.86	0.86	0.86	1089



	precision	recall	f1-score	support
0	0.79	0.86	0.82	537
1	0.85	0.78	0.81	552
accuracy			0.82	1089
macro avg	0.82	0.82	0.82	1089
weighted avg	0.82	0.82	0.82	1089



	precision	recall	f1-score	support
0	0.79	0.86	0.82	537
1	0.85	0.77	0.81	552
accuracy			0.82	1089
macro avg	0.82	0.82	0.82	1089
weighted avg	0.82	0.82	0.82	1089

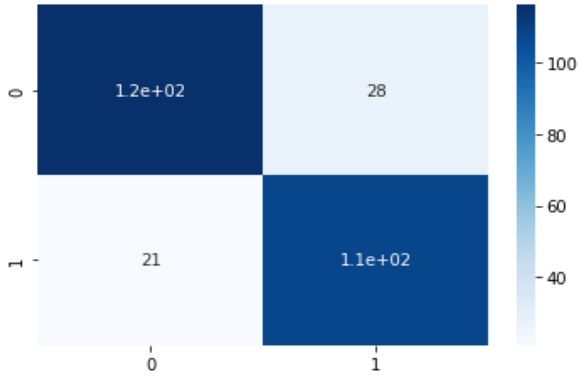


	precision	recall	f1-score	support
0	1.00	1.00	1.00	537
1	1.00	1.00	1.00	552
accuracy			1.00	1089
macro avg	1.00	1.00	1.00	1089
weighted avg	1.00	1.00	1.00	1089

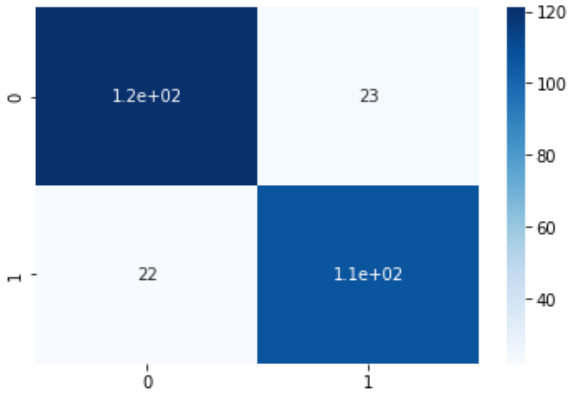
- **Validation du modèle entrainé:**

```
[41] for model in models:
        y_pred=model.predict(x_valid)
        sns.heatmap(confusion_matrix(y_valid,y_pred),annot=True,cmap=plt.cm.Blues)
        plt.show()
        print(classification_report(y_valid,y_pred))
```

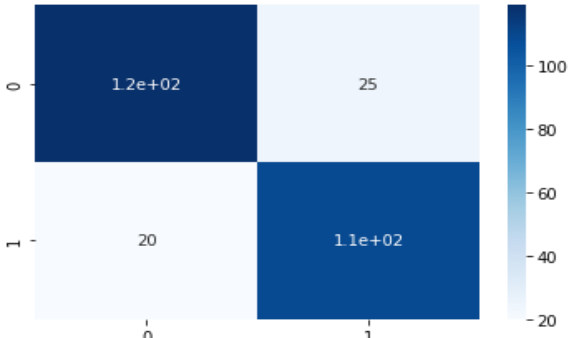
Les résultats de matrice de confusion et de précision pr chaque modèlesont,respectivement, les suivantes :



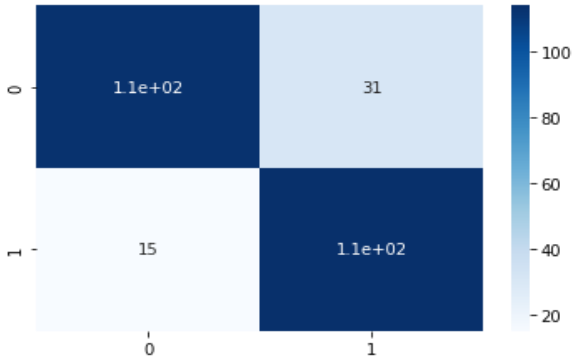
	precision	recall	f1-score	support
0	0.85	0.81	0.83	144
1	0.79	0.84	0.82	129
accuracy			0.82	273
macro avg	0.82	0.82	0.82	273
weighted avg	0.82	0.82	0.82	273



	precision	recall	f1-score	support
0	0.85	0.84	0.84	144
1	0.82	0.83	0.83	129
accuracy			0.84	273
macro avg	0.83	0.83	0.83	273
weighted avg	0.84	0.84	0.84	273



	precision	recall	f1-score	support
0	0.86	0.83	0.84	144
1	0.81	0.84	0.83	129
accuracy			0.84	273
macro avg	0.83	0.84	0.83	273
weighted avg	0.84	0.84	0.84	273



	precision	recall	f1-score	support
0	0.88	0.78	0.83	144
1	0.79	0.88	0.83	129
accuracy			0.83	273
macro avg	0.83	0.83	0.83	273
weighted avg	0.84	0.83	0.83	273

Le modèle de **logistique linéaire** présente la plus grande accuracy parmi les quatres modèles, donc cet algorithme sera choisit pour construire notre modèle

4) Sélection des variables significatives :

Les algorithmes de sélection de fonctionnalités séquentielles sont une famille d'algorithmes de grid search qui sont utilisés pour rérduire la dimension initiale à un sous-espace de fonctionnalités à k dimensions où k < d. La motivation des algorithmes de sélection de fonctionnalités est de sélectionner automatiquement un sous-ensemble de caractéristiques les plus pertinentes pour le problème.

Dans notre cas, on va utiliser Sequential Forward Selection (SFS) à travers la fonction SequentialFeatureSelector :

```
from sklearn.metrics import accuracy_score as acc
from mlxtend.feature_selection import SequentialFeatureSelector as SFS
y_train1 = y_train1.ravel()
y_valid = y_valid.ravel()
sfs1 = SFS(LinearDiscriminantAnalysis(),
           k_features=(10,25),
           forward=True,
           floating=False,
           verbose=2,
           scoring='accuracy',
           cv=10)
  # Perform SFF
sfs1 = sfs1.fit(x_train1, y_train1)
```

On sélectionne les variables choisits par l'application de cet algorithme :

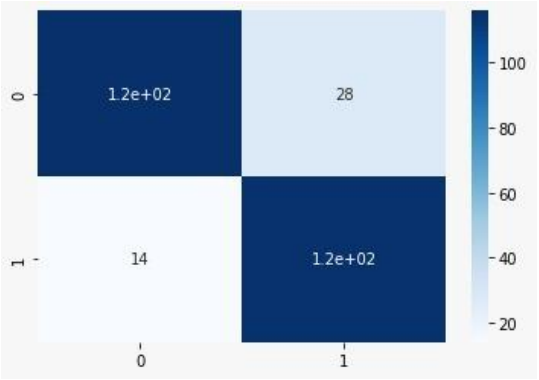
```
feat_cols = list(sfs1.k_feature_idx_)
```

5) Modèle finale :

On traine notre modèle optimale en terme d'accuracy, à l'aide de l'ensemble d'entraînement :

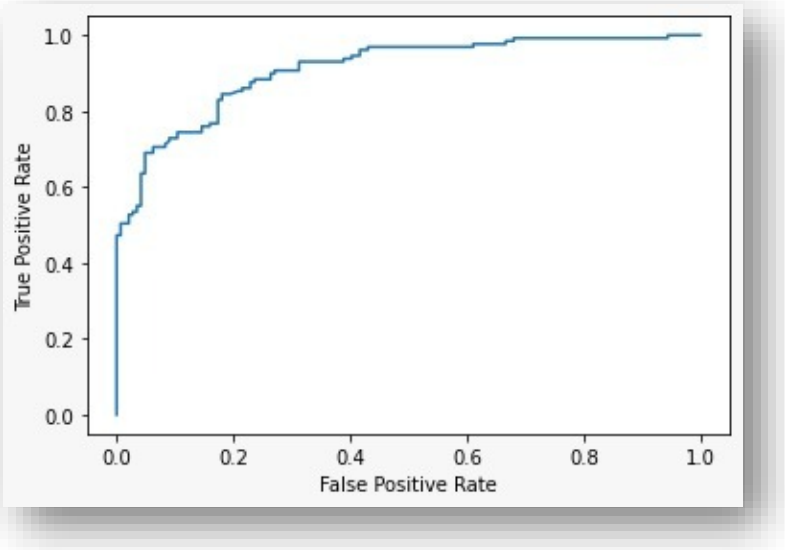
```
y_pred=model.predict(x_test[:,feat_cols])
model.fit(x_train[:,feat_cols],y_train)
sns.heatmap(confusion_matrix(y_test,y_pred),annot=True,cmap=plt.cm.Blues)
plt.show()
print(classification_report(y_test,y_pred))
```

Qui est 85% perfomant pour prédire le risque qui présente chaque client candidat d'un crédit :



	precision	recall	f1-score	support
0	0.89	0.81	0.85	144
1	0.80	0.89	0.85	129
accuracy			0.85	273
macro avg	0.85	0.85	0.85	273
weighted avg	0.85	0.85	0.85	273

6) Courbe ROC :



La courbe ROC est une mesure de la performance du classificateur binaire, elle représente un bon classifieur quand elle s'approche à la courbe de classifieur parfait, ce qui'est le cas pour notre classifieur.

L'aire sous la courbe ROC peut être interprétée comme la probabilité que, parmi deux sujets choisis au hasard, un client ayant un bon crédit et un autre ayant un mauvais crédit, la valeur du marqueur soit plus élevée pour le premier que pour le deuxième. Par conséquent, une AUC de 0,5 (50%) indique que le marqueur est non-informatif. Une augmentation de l'AUC indique une amélioration des capacités discriminatoires, avec un maximum de 1,0 (100%).

Conclusion :

Ce projet présente un outil d'aide a la décision pour les entreprises bancaires à travèrs l'analyse des donnés des clients ayant des crédits bancaire et par la suite la construction d'un modèle d'apprentissage automatique optimale présentant la precision la plus elevée parmi un ensemble d'algorithme afin de classifier le client comme un bon ou mauvais payeur.