## Modèles testés:

### Analyse discriminante linéaire (LDA):

L’analyse discriminante est considérée comme étant parmi les méthodes de classification les plus anciennes.  Elle est basée sur la classification bayésienne en estimant la densité de probabilité.  On peut distinguer deux types de classification selon les hypothèses apliquées sur la matrice de variance covariance:

* Analyse discriminante linéaire
* Analyse discriminante quadratique

**Types de données en entrée:**

* Valeur numérique (Encodage des variables qualitatives)
* Variable normalisée (Suivant une règle de normalisation).
* Aucune valeurs manquantes.
* Transformation des dates (Comptant le nombre de jour depuis une data de référence)

**Avantage:**

* Données: Performe mieux que les autres techniques sur les données de taille inférieure

**Inconvénient:**

* Nécessite la vérification de certaines hypothèses:
  + Variable gaussienne
  + Egalité de la matrice de variance covariance

### Gradient Boosting:

C’est un concept basé sur plusieurs techniques de Machine Learning appelé ”based classifier”. Le but est de trouver un classifieur  F∗ : R  −>  [0,1]  qui assigne la probabilité de défaut d’un client de telle sorte que la moyenne des pertes définis par la fonction de perte L(y,F(x)) soit minimisée pour tous les clients. Cette technique choisit une fonction initiale F0=γ qui est la moyenne en générale et metà jour les paramètres de cette fonction en se basant sur l’erreur commise par le classifieur lors de l’apprentissage. Elle est donc unealternative aux réseaux de neurones artificiels.

**Types de données en entrée:**

* Valeur numérique (Encodage des variables qualitatives)

**Avantage:**

* Données: Performe mieux que les autres techniques sur les données de taille inférieure
* Insensible aux valeurs aberrantes

**Inconvénient:**

* Modele non interpretable

### Forêt aléatoire (Random Forest):

Le principe de cette technique est de combiner plusieurs arbres de décisions. Vu qu’un arbre peut contenir plusieurs règles de décisions, il peut y avoir un ”overfit” du modèle sur les données d'entraînement. Un modèle est en ”overfit” lorsqu’il est trop complexe, il prédit correctement les données d'entraînement mais lors de la phase de test il n’est plus aussi précis. Il existe deux méthodes afin de réduire cet overfit de modèle. La première est l’utilisation d’une technique appelée “Bagging” (Bootstrap aggregating) qui consiste à construire un sous-ensemble aléatoire issu des données principales pour chaque arbre de décision construit. L’autre option consiste à randomiser le sous-ensemble de variables avec lequel chaque arbre est construit. Ces deux méthodes améliorent la capacité deprédiction d’une méthode tout en la stabilisant.

**Types de données en entrée:**

* Valeur numérique (Encodage des variables qualitatives)
* Transformation des dates (Comptant le nombre de jour depuis une data de référence)

**Avantage:**

* Performant avec un large dataset.
* Effectue lui même un feature selection.
* Il peut être adapté aux données contenant des valeurs manquantes

**Inconvénient:**

* Modèle boîte noire
* Interprétation compliquée
* Demande une puissance de calcul conséquente.

### Réseau de neurones (Neural Network):

Le système de réseaux de neurones se base sur le fonctionnement du cerveau humain pour prendre des décisions afin de résoudre des problèmes. Elle met en place des neurones qui sont connectés entre eux et une fonction de perte à minimiser qui mesure l’erreur de classification du modèle pour mettre à jour ses paramètres.

**Types de données en entrée:**

* Valeur numérique (Encodage des variables qualitatives).
* Variable normalisée (Suivant une règle de normalisation).
* Aucune valeurs manquantes.
* Transformation des dates (Comptant le nombre de jour depuis une data de référence)

**Avantage:**

* Algorithme:
  + Inutilité de l’ingénierie des variables
  + Insensible aux valeurs aberrantes
  + Modèle plus souple grâce à la présence des hyperparamètres

**Inconvénients:**

* Modèle compliquée
* Modèle difficilement interprétable
* Temps de calcul élevé

Transformation des variables numériques

Les transformations des variables numériques consistent le plus souvent à un changement d’échelle pour une meilleure lisibilité (modèle interprétable) ou des transformations mathématiques imposées par le modèle (nécessité).

On a effectué une transformation logarithmique dans certains cas. Une échelle logarithmique est un moyen d'afficher des données numériques sur une très large plage de valeurs de manière compacte. Elle permet donc de compacter les variables numériques continues sur une plage de valeurs plus courte (meilleure lisibilité).

Par ailleurs, certains modèles testés nécessitent des variables continues qui suivent une loi normale, d’où la transformation BoxCox qui permet d’approcher la distribution initiale à une distribution normale.

D’autres méthodes existent mais seules les plus pertinentes ont été présentées et testées.

### Transformation des variables catégoriques

Les données catégoriques ne peuvent pas être directement gérées par les modèles mathématiques mais doivent être "numérisées" pour être traitées, c'est-à-dire que les modalités en chaîne de caractère ne sont pas directement exploitables.

Il est donc nécessaire d’identifier et d’analyser chaque variable catégorique car les transformations dépendent du type de ces données catégoriques :

* données de nature nominale (sexe, nationalité, etc) ;
* données catégoriques ordinales (rang, étape, etc).

Nous avons expérimenté les transformations les plus appropriées selon la nature de la variable :

* one hot encoding : pour les variables à faible modalités qui sont  *type de contrat* et *conventionné* avec 2 modalités chacune.

L’encodage one hot consiste à créer une colonne pour chaque modalité et on note 1 si le dossier possède la modalité sinon 0. Dans certains cas de modélisation, l’encodage est utilisé pour toutes les variables catégoriques car il reste le moyen le plus simple et efficace d’exploiter ces variables.

Son seul inconvénient est qu’il crée autant de variables que de modalités. Si une variable possède 20 modalités, il y aura donc 20 nouvelles variables.

* ordinal encoding ou label encoding : adapté pour les variables catégoriques ordinales mais peut s’appliquer à toutes les variables si le modèle n’est pas sensible au poids assigné à chaque modalités.   
  Exemple : “*red*” = 1, “*green*” = 2, and “*blue*” = 3.
* count or frequency encoding : consiste à transformer la modalité par le nombre d'instances de celle-ci ou par sa fréquence par rapport aux données d’apprentissage.
* target encoding : ce procédé transforme chaque valeur par la moyenne ou la probabilité de la valeur cible (target) correspondant à chaque modalité. Cependant, cette méthode favorise le sur-apprentissage car elle est directement liée à la variable cible (ici "état du dossier”).