Université de Paris

Master 2 informatique spécialité Intelligence artificielle – Parcours Apprentissage Machine pour la science des données

Type de projet

---

# Apprentissage supervisé : Détection de la fraude financière à l'aide de l'apprentissage automatique

---

*Auteurs :*
M[lle] Dalia Akhrouf

*Encadrants :*
M Lazhar LABIOD

14 janvier 2020

# Table des matières

# Table des figures

# Introduction

La fraude par carte de crédit est un terme très large pour le vol et la fraude commis en utilisant ou impliquant une carte de paiement, telle qu'une carte de crédit ou une carte de débit, comme source frauduleuse de fonds dans une transaction. Le but peut être d'obtenir des biens sans payer ou d'obtenir des fonds non autorisés à partir d'un compte. La fraude par carte de crédit est également un complément au vol d'identité. Les banques, les commerçants et les sociétés de traitement des cartes de crédit perdent chaque année des milliards de dollars à cause de la fraude par carte de crédit.

## 0.1 Objectifs du projet :

L'objectif de ce travail est la mise en pratique concrète d'un certain nombre de techniques d'apprentissage supervisé (Bayesien Naïf, KNN, LDA, QDA, Linear SVM, Non Linear SVM, Régression logistique, CART et Random Forest), à travers l'étude de données synthétiques et deux cas pratiques.
La détection de la fraude par carte de crédit a deux objectifs. Il aide les commerçants et les banques à réduire le nombre de cas de fraude de paiement et aide les commerçants à augmenter leurs revenus. Les sociétés de cartes de crédit détectent la fraude en signalant plusieurs types de transactions. Parmi eux, les achats importants effectués juste après les petits, les achats en ligne et les achats qui ne correspondent pas au profil du titulaire de carte.

# Chapitre 1

# Etude sur données synthétiques

## 1.1 Objectif de cette partie :

Cette partie concerne un travail sur données synthétiques. Il s'agit de mettre en oeuvre des méthodes citées dans l'introduction sur l'ensemble des jeux de données synthétiques proposés. Le travail consiste à réaliser une étude comparative des ces différentes approches de classification supervisée.

## 1.2 La description des données

Il s'agit de 3 bases de données synthétiques possédant des caractéristiques différentes, en termes de nombre de classes et de la structure des classes.

| Tables | Nombre d'observations | Nombre de variables | Nombre de classes |
|---|---|---|---|
| Flame | 240 | 2 | 2 |
| Spiral | 312 | 2 | 3 |
| Aggregation | 788 | 2 | 7 |

TABLE 1.1 – Caractéristiques des bases de données.

## 1.3 Étude de la base de données Flame

### 1.3.1 accuracy erreur :

Train data : knn : accuaracy= 0.99 erreur= 0.01 , LDA : accuaracy= 0.86 erreur= 0.14, rendom forest : accuaracy= 1.0 erreur= 0.0, decision tree : accuaracy= 1.0 erreur= 0.0, linear SVM : accuaracy= 0.86 erreur= 0.14, Guasian SVM : accuaracy= 0.99 erreur= 0.01, logistic regression : accuaracy= 0.85 erreur= 0.15, naive bayes : accuaracy= 0.95 erreur= 0.05, QDA : accuaracy= 0.96 erreur= 0.04

test Data : knn : accuaracy= 1.0 erreur= 0.0, LDA : accuaracy= 0.94 erreur= 0.06, rendom forest : accuaracy= 1.0 erreur= 0.0, decision tree : accuaracy= 1.0 erreur= 0.0, linear SVM : accuaracy= 0.93 erreur= 0.07, Guasian SVM : accuaracy= 1.0 erreur= 0.0, logistic regression : accuaracy= 0.96 erreur= 0.04, naive bayes : accuaracy= 0.99 erreur= 0.01, QDA : accuaracy= 1.0 erreur= 0.0

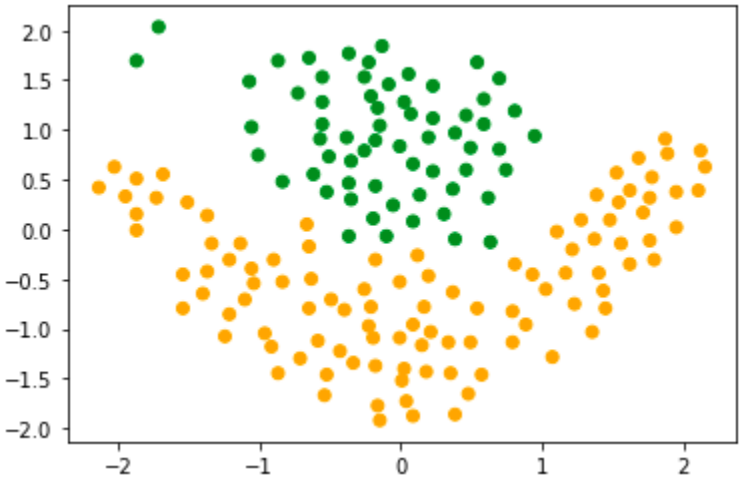


FIGURE 1.1 – flame train Data

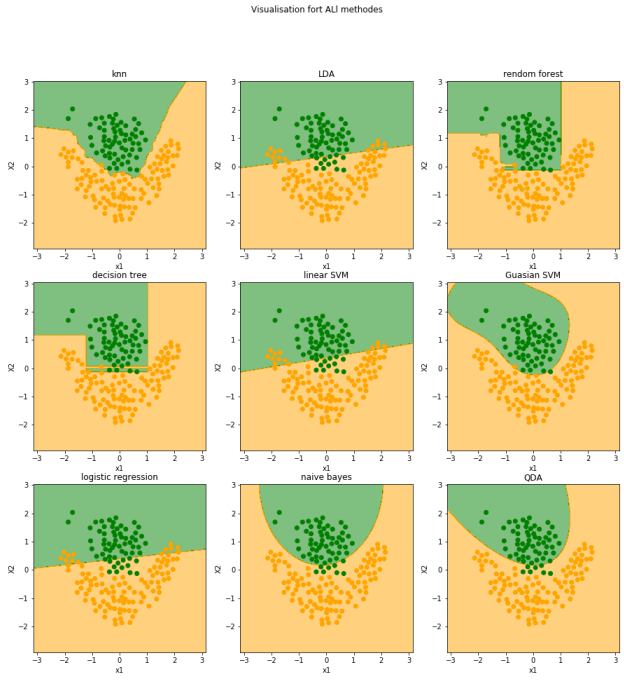



FIGURE 1.2 – flame train data with all modles

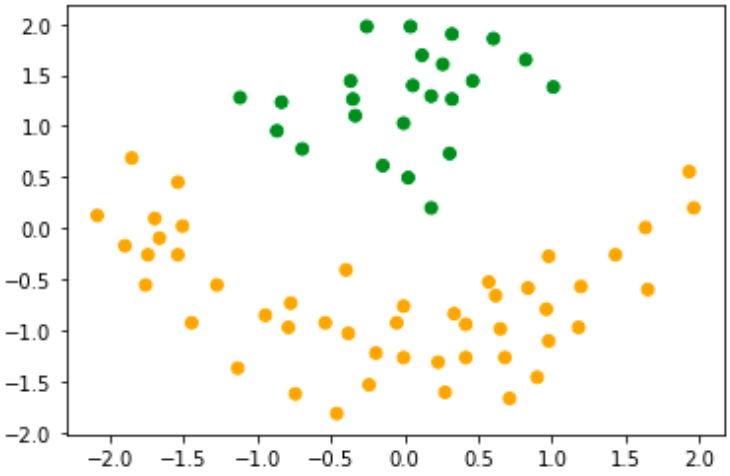


FIGURE 1.3 – flame test data

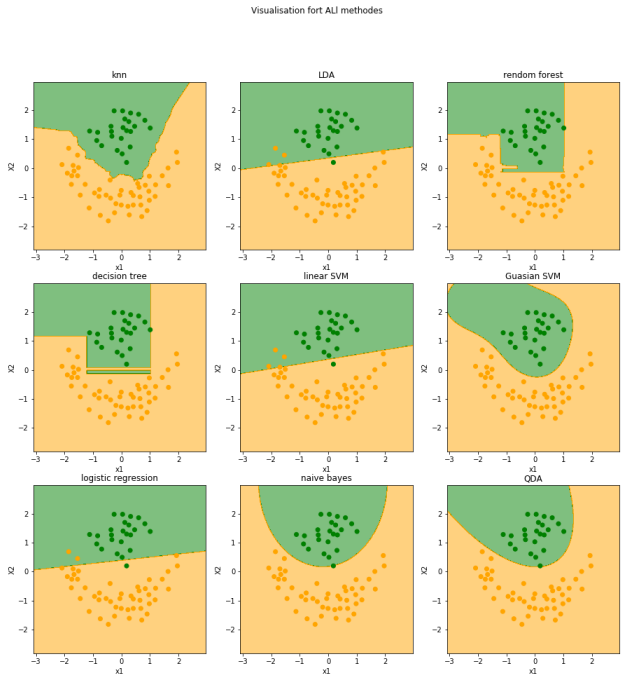



FIGURE 1.4 – flame test data with all models

## 1.4 Étude de la base de données Spiral

train data : knn : accuaracy= 1.0 erreur= 0.0, LDA : accuaracy= 0.36 erreur= 0.64, rendom forest : accuaracy= 1.0 erreur= 0.0, decision tree : accuaracy= 1.0 erreur= 0.0, linear SVM : accuaracy= 0.35 erreur= 0.65, Guasian SVM : accuaracy= 0.98 erreur= 0.02, logistic regression : accuaracy= 0.37 erreur= 0.63, naive bayes : accuaracy= 0.37 erreur= 0.63, QDA : accuaracy=

0.38 erreur= 0.62

test data : knn : accuaracy= 0.99 erreur= 0.01, LDA : accuaracy= 0.34 erreur= 0.66, rendom forest : accuaracy= 0.96 erreur= 0.04, decision tree : accuaracy= 0.99 erreur= 0.01, linear SVM : accuaracy= 0.33 erreur= 0.67, Guasian SVM : accuaracy= 0.93 erreur= 0.07, logistic regression : accuaracy= 0.34 erreur= 0.66, naive bayes : accuaracy= 0.34 erreur= 0.66, QDA : accuaracy= 0.32 erreur= 0.68

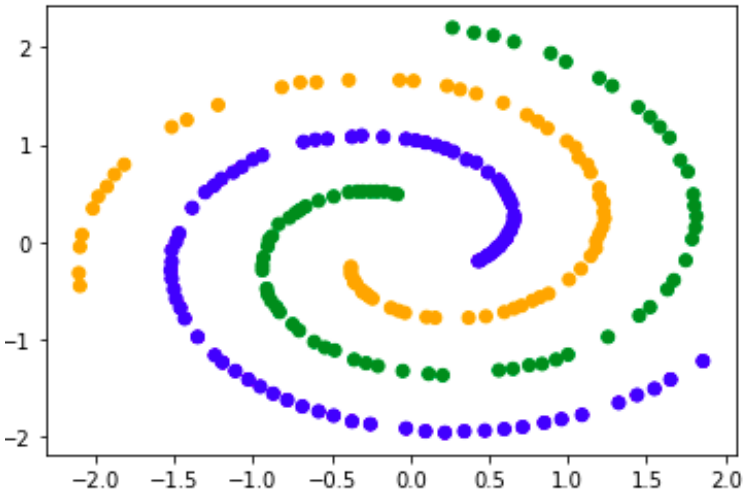



FIGURE 1.5 – Spiral train Data

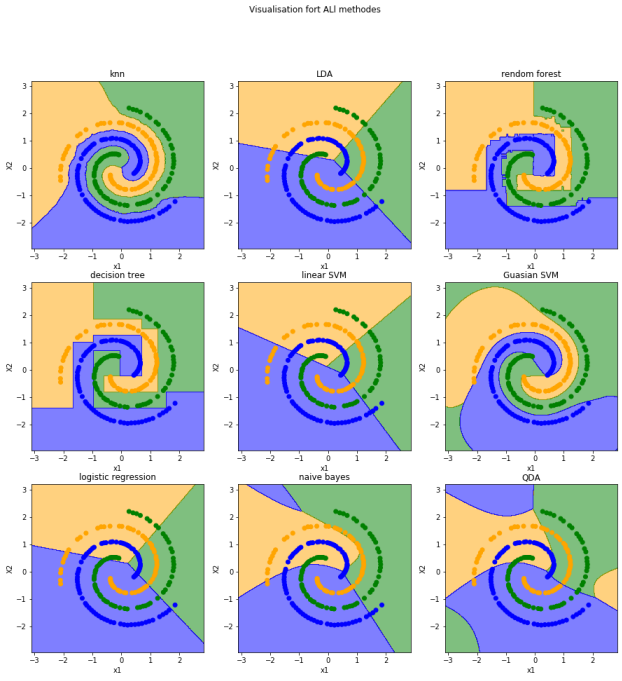
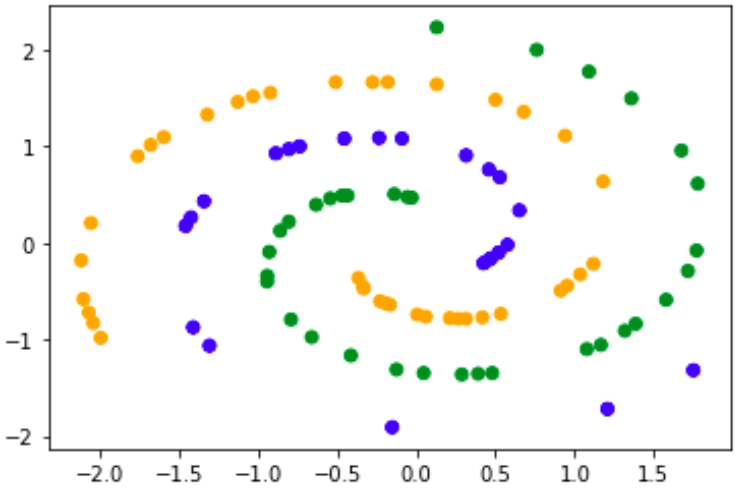



FIGURE 1.6 – Spiral train data with all models



$_t rain.png$

FIGURE 1.7 – Spiral test data

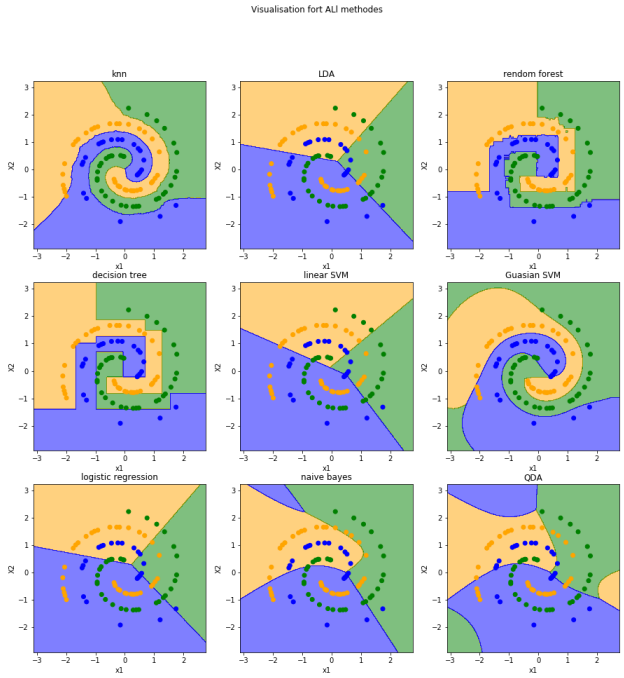



FIGURE 1.8 – Spiral test data with all models

## 1.5 Étude de la base de données Aggregation

Train Data : knn : accuaracy= 1.0 erreur= 0.0, LDA : accuaracy= 1.0 erreur= 0.0, rendom forest : accuaracy= 1.0 erreur= 0.0, decision tree : accuaracy= 1.0 erreur= 0.0, linear SVM : accuaracy= 1.0 erreur= 0.0, Guasian SVM : accuaracy= 1.0 erreur= 0.0, logistic regression :

accuaracy= 0.87 erreur= 0.13, naive bayes : accuaracy= 1.0 erreur= 0.0, QDA : accuaracy= 1.0 erreur= 0.0,

Test data : knn : accuaracy= 1.0 erreur= 0.0, LDA : accuaracy= 0.99 erreur= 0.01, rendom forest : accuaracy= 0.98 erreur= 0.02, decision tree : accuaracy= 1.0 erreur= 0.0, linear SVM : accuaracy= 1.0 erreur= 0.0, Guasian SVM : accuaracy= 1.0 erreur= 0.0, logistic regression : accuaracy= 0.82 erreur= 0.18, naive bayes : accuaracy= 1.0 erreur= 0.0, QDA : accuaracy= 1.0 erreur= 0.0,

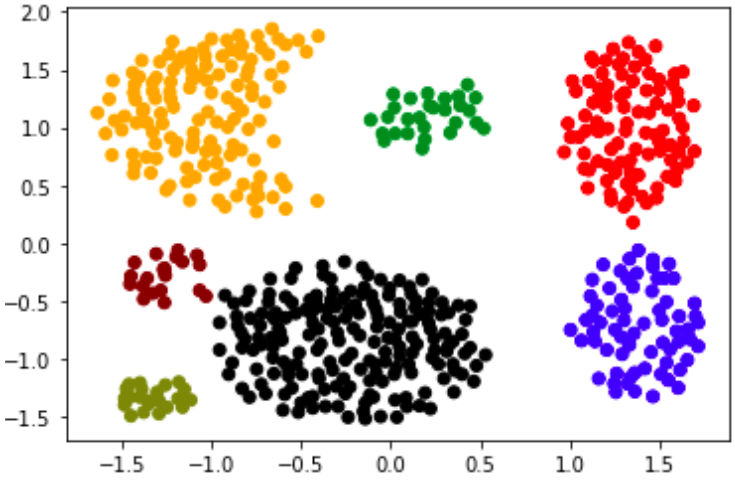



FIGURE 1.9 – Agregation train data

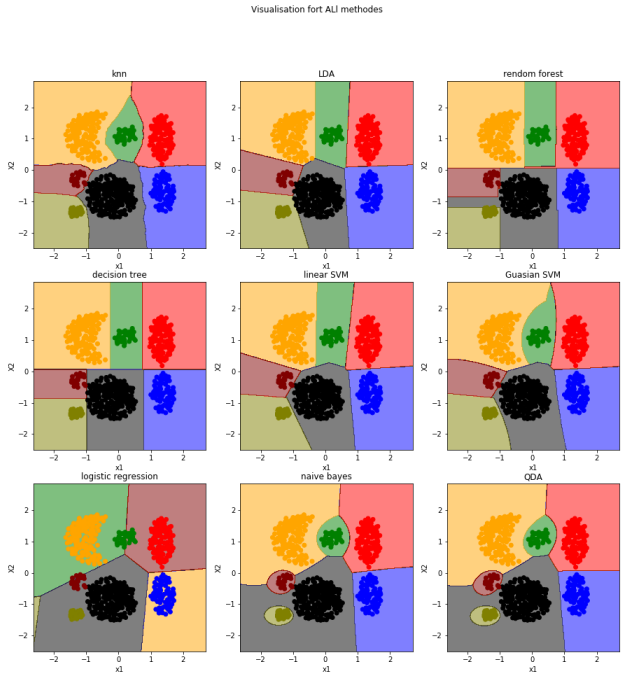



FIGURE 1.10 – Agregation train data with all models

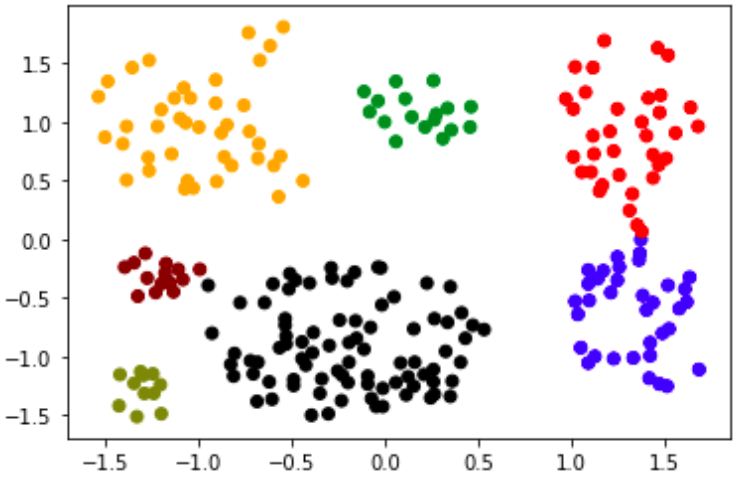


FIGURE 1.11 – Agregation test Data

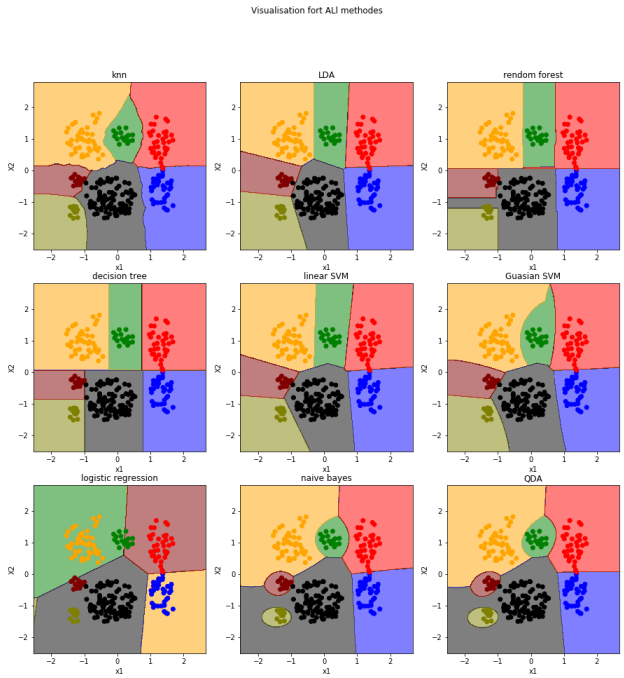



FIGURE 1.12 – Agregation test Data with all models

## 1.6 Résultats et Discussion

la comparaison des résultats est résumé dans les tableaux suivants :

| | KNN | LDA | R.Forest | D.Tree | L.SVM | G.SVM | L. regression | NB | QDA |
|---|---|---|---|---|---|---|---|---|---|
| Accuaracy | 0.99 | 0.86 | 1.0 | 1.0 | 0.86 | 0.99 | 0.85 | 0.95 | 0.96 |
| erreur | 0.01 | 0.14 | 0.0 | 0.0 | 0.14 | 0.01 | 0.15 | 0.05 | 0.04 |
| Accuaracy | 1.0 | 0.36 | 1.0 | 1.0 | 0.35 | 0.98 | 0.37 | 0.37 | 0.38 |
| erreur | 0.0 | 0.64 | 0.0 | 0.0 | 0.65 | 0.02 | 0.63 | 0.63 | 0.62 |
| Accuaracy | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 0.87 | 1.0 | 1.0 |
| erreur | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.13 | 0.0 | 0.0 |

TABLE 1.2 – Résultats obtenus pour les données Train.

| | KNN | LDA | R.Forest | D.Tree | L.SVM | G.SVM | L. regression | NB | QDA |
|---|---|---|---|---|---|---|---|---|---|
| Accuaracy | 1.0 | 0.94 | 1.0 | 1.0 | 0.93 | 1.0 | 0.96 | 0.99 | 1.0 |
| erreur | 0.0 | 0.06 | 0.0 | 0.0 | 0.07 | 0.0 | 0.04 | 0.01 | 0.0 |
| Accuaracy | 0.99 | 0.34 | 0.96 | 0.99 | 0.33 | 0.93 | 0.34 | 0.34 | 0.32 |
| erreur | 0.01 | 0.66 | 0.04 | 0.01 | 0.67 | 0.07 | 0.66 | 0.66 | 0.68 |
| Accuaracy | 1.0 | 0.99 | 0.98 | 1.0 | 1.0 | 1.0 | 0.82 | 1.0 | 1.0 |
| erreur | 0.0 | 0.01 | 0.02 | 0.0 | 0.0 | 0.0 | 0.18 | 0.0 | 0.0 |

TABLE 1.3 – Résultats obtenus pour les données Test.

# Chapitre 2

# Étude de cas pratiques

## 2.1 Objectif de cette partie :

Cette partie s'intéresse à deux cas pratiques (clients d'une banque, transactions bancaires), l'objectif est d'appliquer les différentes approches citées dans l'introduction, choisir pour chaque méthode le meilleur modèle et ensuite comparer ces modèles sur un ensemble de test qui n'a pas été utilisé dans les phases d'apprentissage et de validation des modèles en concurrence.

## 2.2 Analyse exploratoire des données "Credit card Fraud"

### 2.2.1 La description des données

| Nom de l'attribut | Explication |
|---|---|
| 1.Time | L'attribut 'Time' contient les secondes écoulées entre chaque transaction et la première transaction de l'ensemble de données. Cet ensemble de données présente les transactions qui ont eu lieu en deux jours |
| 2.PCA Variables (V1,V2,,V28) | L'ensemble de données contient uniquement des variables d'entrée numériques qui sont le résultat d'une transformation PCA (analyse en composantes principales). Malheureusement, en raison de problèmes de confidentialité, les fonctionnalités d'origine et plus d'informations de fond sur les données ne sont pas fournies. Caractéristiques V1, V2, ...V28 sont les principaux composants obtenus avec PCA |
| 3. Amount | L'attribut 'Amount' est le montant de la transaction, cette fonctionnalité peut être utilisée pour un apprentissage sensible aux coûts dépendant de l'exemple. Le montant d'argent retiré par quelqu'un dans une transaction est simplement appelé ici « Amount ». |
| 4.Class | The feature 'Class' est la variable de réponse et prend la valeur 1 en cas de fraude et 0 sinon. Nous avons 492 fraudes sur 284 807 transactions. L'ensemble de données est très déséquilibré, la classe positive (fraudes) représente 0,172% de toutes les transactions. |

Table 2.1 – Caractéristiques de la base des données.

-Nous commencerons par une analyse exploratoire du jeu de données avec la particularité que nos données ne sont pas équilibrées, c'est-à-dire que seulement 0,17% du total de la transaction par carte de crédit est frauduleux.
-un déséquilibre de classe est un problème très courant dans la vie réelle et doit être traité avant de lui appliquer un algorithme. Il existe trois façons courantes de remédier au déséquilibre des données :

— Sous échantillonnage Échantillonnage unilatéral par Kubat et Matwin (ICML 1997)
— Sur échantillonnage frappèrent (Technique synthétique Minority suréchantillonnage)
— Combiner les deux ci-dessus.

Notre jeux de données a subi une ACP, donc les variables devraient être linéairement décorélées.

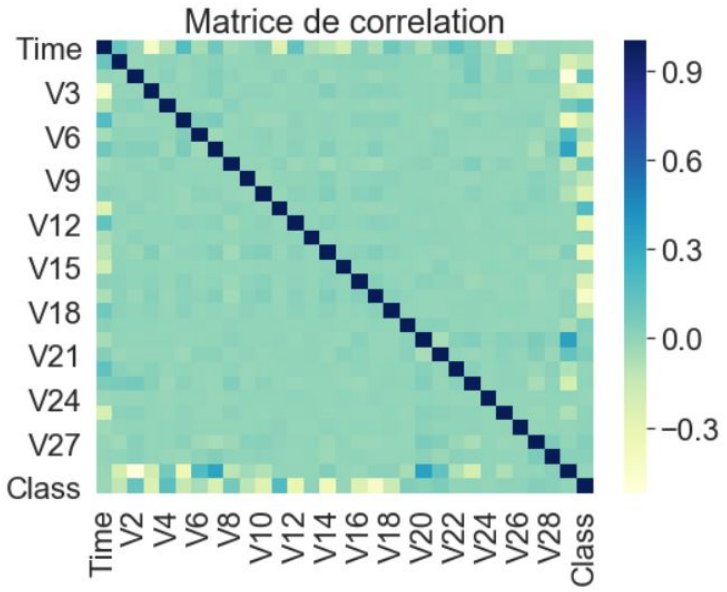



FIGURE 2.1 – Matrice de corrélation

Notre jeux de données ne présente pas de problème de multicolinéarité donc nous aurons pas de soucis avec la régression logistique et l'analyse discriminante (linéaire et quadratique) pour la suite.

Séparons les cas frauduleux des cas authentiques et comparons leurs occurrences dans l'ensemble de données.

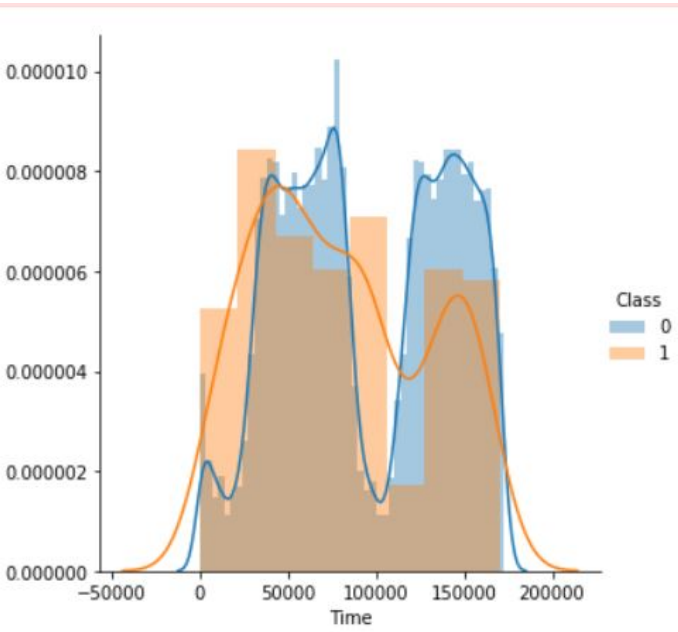



FIGURE 2.2 – fraud Vs normal

Observations :

Il y a un chevauchement important des transactions normales et frauduleuses tout au long du temps et il n'y a pas de distinction claire.

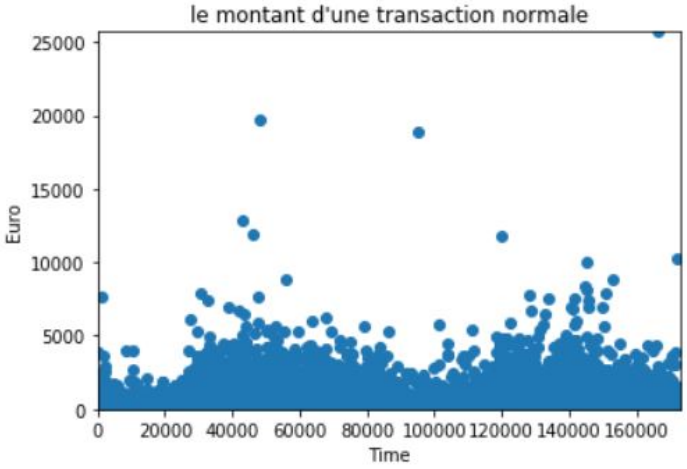



FIGURE 2.3 – montant des transactions normales

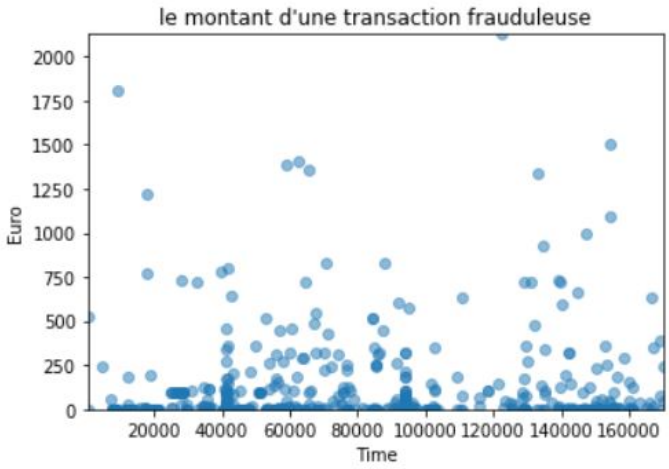



FIGURE 2.4 – montant des transactions frauduleuses

Observations :

D'après les deux graphiques ci-dessus, il est clairement visible qu'il n'y a de fraude que sur les transactions dont le montant de la transaction est inférieur à environ 2500. Les transactions dont le montant de la transaction est supérieur à 2500 environ n'ont pas de fraude. Selon le temps, Il ne semble pas que le moment de la transaction soit vraiment important, les fraudes dans les transactions sont également réparties dans le temps.

### 2.2.2 Traitements des données :

L'ensemble de données n'est pas équilibré et avec des algorithmes d'apprentissage ils pouvent conduire à une classification erronée de la classe minoritaire. Par conséquent, pour compenser le déséquilibre, nous utiliserons la méthode de suréchantillonnage ADASYN telle qu'implémentée dans le package d'apprentissage déséquilibré pour rééchantillonner l'ensemble de données. ADASYN (ADAptive SYNthetic) est une technique de suréchantillonnage qui génère de manière adaptative des échantillons de données minoritaires en fonction de leurs distributions en utilisant K le plus proche voisin.
Notre nouveau jeu de données contient : dans la classe 0 : 19896 transactions et dans la classe 1 : 19880 transactions.
Dans notre cas, nous n'utilisons pas de sous-échantillonnage aléatoire puisque cela peut poser des problèmes d'informations utiles sur les données elles-mêmes, qui pourraient s'avérer nécessaires pour créer des classifier basés sur des règles tels que Random Forest ou le Gradient Boosting.

### 2.2.3 Modèle d'apprentissage supervisé

**Classification naïve bayésienne :**

Le classifieur naïf bayésien est l'une des méthodes les plus simples en apprentissage supervisé et il est basé sur le théorème de Bayes. Cet algorithme s'appelle «naïf» car il suppose naïvement que chaque caractéristique est indépendante des autres, ce qui est faux dans la vie réelle.
Ensuite le théorème de Bayes nous aide à trouver la probabilité d'une hypothèse compte tenu de nos connaissances antérieures.

Cross Validation Mean Score : 94.19999999999999%
Model Accuracy : 94.3

**Régression Logistique :**

Le modèle de régression logistique prend des entrées à valeur réelle et permet de prédire la probabilité que l'entrée appartienne à une classe. Ainsi si la probabilité est supérieur à 0.5, nous pouvons prendre le résultat comme une prédiction pour la classe transaction frauduleuse, et sinon la prédiction est pour l'autre classe (transaction non frauduleuse).
Cross Validation Mean Score : 96.89999999999999%
Model Accuracy : 95.8

**Random Forest :**

Les forêts aléatoires sont des modèles puissant et précis puis ils sont performant sur de nombreux problèmes non-linéaire. En revanche, ils ne sont pas facilement interprétables et le nombre d'arbres choisis est importants car il peut y avoir des problèmes d'over-fitting.
Cross Validation Mean Score : 99.6%
Model Accuracy : 100.0

**Comparaison des résultats obtenus entre les models :**

```
============================== ROC Curve ==================================
```

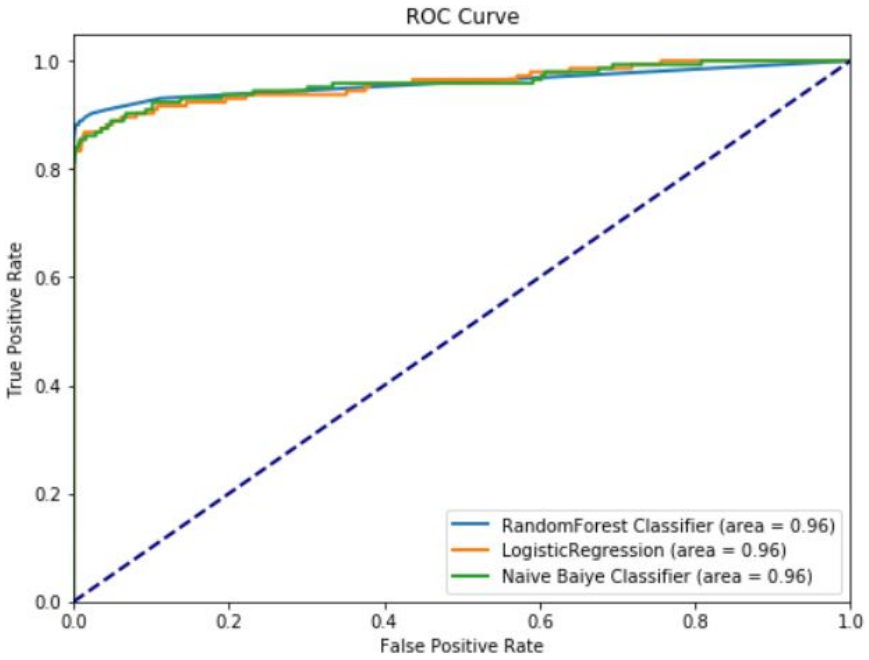



Figure 2.5 – ROC Curve

**Analyse Linéaire et Quadratique Dicriminante QDA :**

L'analyse discriminante linéaire (ADL) recherche la projection qui minimise la variance intra-classe de cet ensemble de données projeté ainsi que de maximiser la variance inter-classe. Et de plus, l'analyse discriminante linéaire a pour objectif de projeter les points situées dans un espace de dimension supérieure sur un espace de dimension inférieure.

**Arbre de décision :**

Un arbre de décision est un arbre dans lequel chaque noeud représente une caractéristique, chaque branche représente une décision et chaque feuille représente un résultat (catégoriel ou continue). L'idée est de créer un tel arbre pour l'ensemble des données et de traiter un résultat unique à chaque feuille (ou de minimiser l'erreur dans chaque feuille).

| | MLA Name | Precision | AUC |
|---|---|---|---|
| 1 | QuadraticDiscriminantAnalysis | 0.054112 | 0.927959 |
| 0 | DecisionTreeClassifier | 0.087881 | 0.923012 |

FIGURE 2.6 – Comparaison entre les arbres de decision et QDA

**Machine à vecteurs de support (SVM) :**

La SVM consiste à projeter d'abord les données sur un espace de dimensions plus grandes, où nous pouvons encapsuler les données normales dans une hypersphère, même si aucune hypersphère ne peut capturer toutes les données de l'espace d'origine.
Nous avons detecter 10 frauds / 16 total des fraudes.
Donc, la probabilité que la fraude soit detecter est : 0.625
la précision est de : 0.9989467524868344

### 2.2.4 Conclusion et Discussion :

En combinant les classificateurs, nous pouvons tirer le meilleur parti de plusieurs modèles. Random Forest en tant que modèle autonome était bon en précision mais assez mauvais en termes de faux négatifs. La régression logistique était bonne dans le rappel, mais très mauvaise en termes de faux positifs. L'arbre de décision était au milieu. En combinant ces modèles, nous avons en effet réussi à améliorer les performances. Nous avons augmenté le nombre de cas de fraude de 75 à 78, et réduit les faux négatifs de 3, et nous n'avons que 4 faux positifs supplémentaires en retour. Si nous nous soucions d'attraper autant de cas de fraude que possible, tout en gardant les faux positifs bas, c'est un très bon compromis.

## 2.3 Analyse exploratoire des données "Visa Premier)" :

Il s'agit d'une base de données décrivant les clients d'une banque et leurs comportements (mouvements, soldes des différents comptes). La variable à expliquer Y est la variable binaire «

Possession de la carte Visa Premier ».

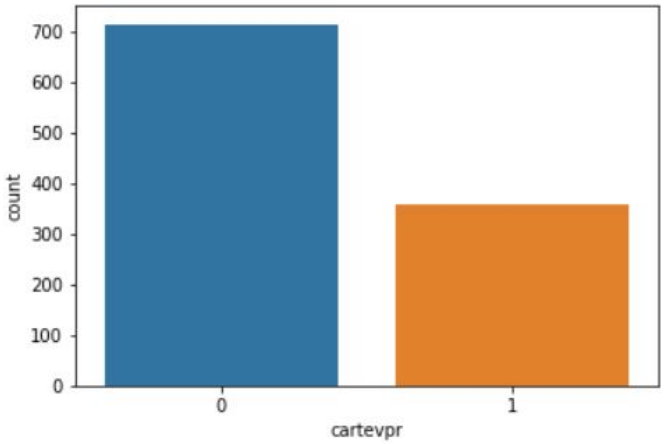



FIGURE 2.7 – repartition des individus selon la procession ou non de la carte

- On remarque que les classes sont déséquilibrées et que la plupart ne procèdent pas de carte.

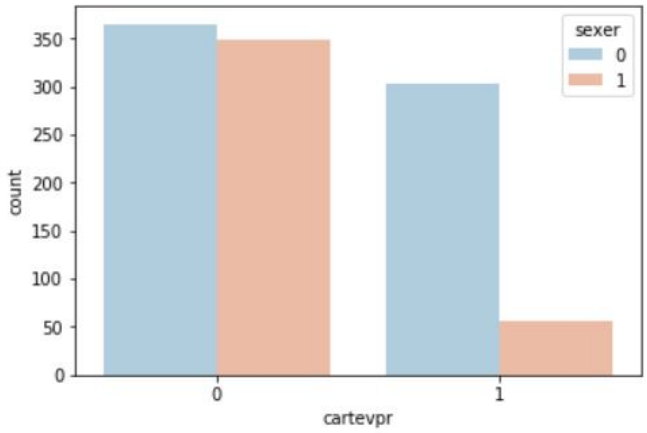



FIGURE 2.8 – répartition des individus selon le sexe

- Après l'étude des données nous avons remarqué que les variables sexe et âge , matricule et les données qualitatives n'influence pas sur la procession ou pas de la carte, c'est pour ce la pour la suite nous avons gardé que les variables essentielles.

### 2.3.1 Modèle d'apprentissage supervisé

| | R.Forest | L.Regression | NB | KNeighbors | LDA | D.Tree | SVM | QDA |
|---|---|---|---|---|---|---|---|---|
| Accuaracy | 98.8% | 86.7% | 77.2% | 98.8% | 85.0% | 100% | 100% | 77.9% |
| Area Under a Curve | 0.89 | 0.91 | 0.81 | 0.89 | 0.90 | 0.76 | 0.93 | 0.82 |

TABLE 2.2 – Résultats obtenus apres l'application des models.

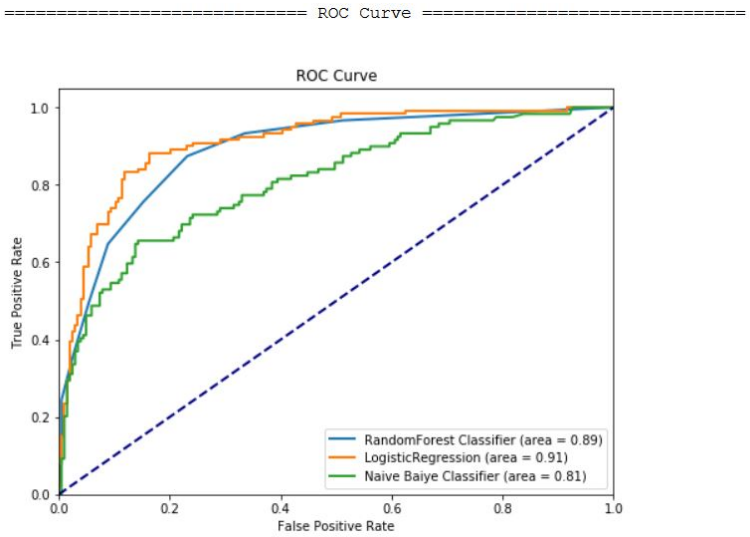



FIGURE 2.9 – La courbe roc pour les models RandomForest LogisticRegression et Naive Baiyes

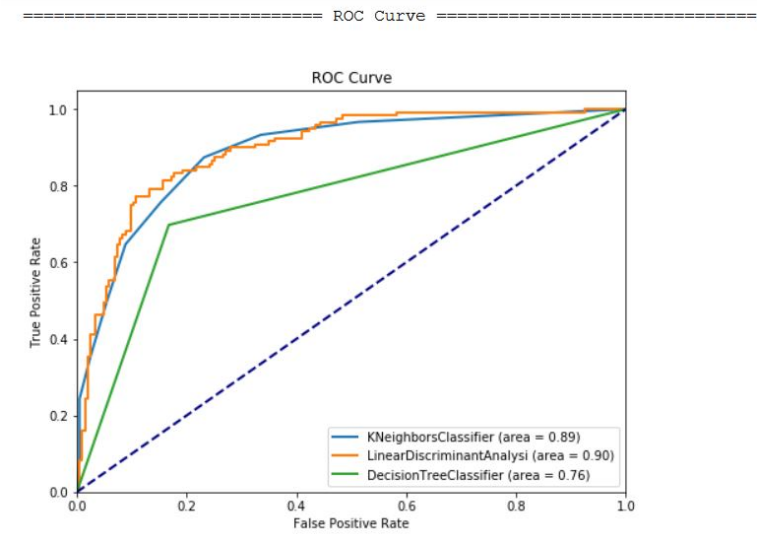



FIGURE 2.10 – La courbe roc pour les models KNeighbor LDA et DecisionTree

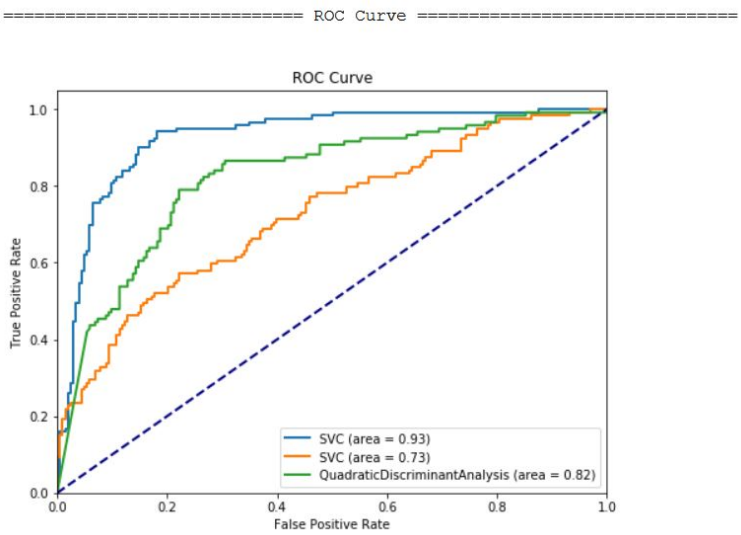



FIGURE 2.11 – La courbe roc pour les models SVM et QDA

### 2.3.2 Conclusion et Discussion

-D'après les résultats, nous avons remarqué les résultats de SVM donne de meilleurs résultats.

## 2.4 Conclusion

Dans ce projet nous avons pu manipuler plusieurs algorithmes de Machine Learning Supervisé.Ce projet nous à permis de pouvoir mettre en pratique les notions vu pendant le cours. Nous les avons utiliser pour de la classification sur des données synthétiques et réelles.

La principale chose que nous avons remarqué entre les 2 types de données, c'est que pour les données synthétiques nous n'avons pas besoin d'équilibrer et de nettoyer les données. On ob- tient donc des résultats excellent avec plusieurs algorithmes alors que dans la réalité il faut extraire les variables les plus importantes et traiter les algorithmes pour pouvoir en tirer le meilleur de chaque.

Pour finir, et nous a permis de nous rendre compte que le travail d'un Data Scientist ne se résume pas au simple fait d'appliquer des algorithmes