



M2 ISIFAR

ÉTUDE SUR LA
DÉTECTION DE FAUSSES MONNAIES

MÉHEUST WILLIAM

RAGAVAN RANUSHAN

Dans le cadre de l'UE Apprentissage Statistique

Sous la direction de Aurelie FISCHER

Année universitaire 2022-2023

Table des matières

1	Introduction	2
2	Analyse descriptive	3
2.1	Étude des valeurs manquantes	3
2.1.1	Analyse des valeurs manquantes	3
2.1.2	Corrélation entre les variables	3
2.1.3	Régression linéaire	4
2.2	Visualisation du jeux de données	6
2.2.1	Utilisation de Pairplot	6
2.2.2	Utilisation de Boxplot	7
2.3	Rééchantillonnage	9
3	Modèles de classifications	10
3.1	Régression logistique	10
3.1.1	Métriques d'évaluation	10
3.2	Arbre de classification	11
3.2.1	Métriques d'évaluation	11
3.3	SVM	13
3.3.1	Hyperparamètres / Cross-validation	13
3.3.2	Métriques d'évaluation	14
3.4	ACP	15
3.5	Forêts aléatoires	16
3.5.1	Hyperparamètres / Cross-validation	16
3.5.2	Métriques d'évaluation	16
3.6	KNN	17
3.6.1	Hyperparamètres / Cross-validation	17
3.6.2	Métriques d'évaluation	17
4	Conclusion	18



1 Introduction

Notre étude statistique portera sur l'analyse et la détection de fausses monnaies. Tout l'enjeu de ce devoir sera donc d'être capable de proposer des méthodes statistiques efficaces dans la détection de fausse monnaie.

Ainsi nous nous sommes procuré un jeu de données (sur KAGGLE) intitulé *fakebills.csv*. Le dataset est constitué de 7 colonnes et 1500 lignes. Nous avons donc 7 variables et 1500 individus, les variables étant pour 6 d'entre elles quantitatives et la variable restante étant qualitative binaire.

Les différentes vraiables correspondent à :

- *is_genuine* : Le billet est-il authentique ? VRAI / FAUX
- *diagonale* : les mesures diagonales en mm
- *height_left* : la hauteur du côté gauche en mm
- *height_right* : la hauteur du côté droit en mm
- *margin_low* : la marge inférieure en mm
- *margin_up* : la marge supérieure en mm
- *longueur* : la longueur en mm

Dans ce dataset chaque billets est donc caractérisé uniquement par les différentes dimensions qui le composent.

L'objectif de cette étude sera donc dans un premier temps d'analyser la base de données et ainsi de déterminer les variables les plus pertinentes, celles qui ont le plus d'impact. Dans un second temps, nous mettrons en pratique différentes méthodes de prédiction étudiée en cours afin de tenter de déterminer si un billet est un faux ou un vrai. Nous mettrons en exerbe la ou les méthodes les plus performantes et expliquerons d'où viennent ces différences.



2 Analyse descriptive

2.1 Étude des valeurs manquantes

2.1.1 Analyse des valeurs manquantes

Nous allons déterminer le nombre de valeurs manquantes dans l'ensemble du dataset. Après analyse nous trouvons 37 valeurs manquantes. Il faut ensuite déterminer leurs localisations. Nous remarquons que les 37 valeurs manquantes sont toutes présentes au même endroit, elles sont toutes dans la variable *margin_low*.

Il existe plusieurs méthodes pour remplacer les données manquantes :

- Remplacer par la moyenne
- Remplacer par la medianne
- Remplacer par la catégorie dominante
- Créer une nouvelle catégorie
- Faire de la régression linéaire
- Faire du k-mean

Nous allons utiliser dans la suite de l'étude la méthode de la régression linéaire pour remplacer les données manquantes, car cette méthode est plus poussée que les méthodes ou nous remplaçons par la moyenne ou par la médiane. De plus nous avons à faire à des variables numériques ainsi utiliser la méthode de la régression linéaire est une option intéressante.

2.1.2 Corrélation entre les variables

Nous voulions également mieux comprendre les corrélations qui liaient les différentes variables, ainsi, nous obtenons cette matrice de corrélation.

	<i>is_genuine</i>	<i>diagonal</i>	<i>height_left</i>	<i>height_right</i>	<i>margin_low</i>
<i>is_genuine</i>	1.0000000	0.13275633	-0.37983292	-0.48509183	NA
<i>diagonal</i>	0.1327563	1.00000000	0.01947232	-0.02449201	NA
<i>height_left</i>	-0.3798329	0.01947232	1.00000000	0.24227881	NA
<i>height_right</i>	-0.4850918	-0.02449201	0.24227881	1.00000000	NA
<i>margin_low</i>	NA	NA	NA	NA	1
<i>margin_up</i>	-0.6062623	-0.05564888	0.24652224	0.30700464	NA
<i>length</i>	0.8492846	0.09758729	-0.32086276	-0.40175122	NA
		<i>margin_up</i>	<i>length</i>		
<i>is_genuine</i>	-0.60626226	0.84928463			
<i>diagonal</i>	-0.05564888	0.09758729			
<i>height_left</i>	0.24652224	-0.32086276			
<i>height_right</i>	0.30700464	-0.40175122			
<i>margin_low</i>		NA	NA		
<i>margin_up</i>		1.00000000	-0.52057513		
<i>length</i>		-0.52057513	1.00000000		

FIG. 1 – Corrélation

Nous ne pouvons donc pas conclure sur les les corrélation linéaire entre *margin_low* et les autres variables, ceci est du aux valeurs manquantes.

Cependant nous remarquons que *is_genuine* est fortement corrélé avec *length* et fortement anticorrélé avec *margin_up*.



2.1.3 Régression linéaire

Comme vu précédemment l'une des variables quantitative possède des valeurs manquantes. Nous allons donc utiliser une méthode de computation via GLM pour les remplacer.

Nous commençons par isoler les données manquantes, puis nous retirons la variable target pour éviter d'utiliser les variables target pour l'imputation et biaisé la suite de l'étude.

Pour effectuer l'imputaiton via GLM il est primordial de déterminer les bonnes varibales a utiliser, pour cela nous allons utiliser les critères AIC et BIC.

- Étude du critère AIC

```

Start: AIC=2955.87          Step: AIC=2097.64          Step: AIC=2052.37          Step: AIC=2035.59
margin_low ~ 1               margin_low ~ length        margin_low ~ length + height_right    margin_low ~ length + height_right + margin_up
                               Df Deviance   AIC           Df Deviance   AIC           Df Deviance   AIC           Df Deviance   AIC
+ length                     1   357.83 2097.6 + height_right  1   346.45 2052.4 + margin_up     1   342.03 2035.6 + height_left  1   338.37 2021.8
+ margin_up                  1   524.22 2656.3 + margin_up     1   351.57 2073.8 + height_left  1   342.03 2035.9 + diagonal    1   340.62 2031.5
+ height_right                1   545.69 2715.0 + height_left  1   351.64 2074.1 + diagonal    1   344.99 2048.2 <none>          1   342.03 2035.6
+ height_left                 1   585.22 2817.3 + diagonal    1   356.55 2094.4 <none>          1   346.45 2052.4 - margin_up    1   346.45 2052.4
+ diagonal                   1   636.21 2939.6 <none>          1   357.83 2097.6 - height_right  1   357.83 2097.6 - height_right  1   351.57 2073.8
<none>                      1   644.23 2955.9 - length       1   644.23 2955.9 - length       1   545.69 2715.0 - length       1   476.64 2519.1
                                         Step: AIC=2016.65
                                         margin_low ~ length + height_right + margin_up + height_left +
margin_low ~ length + height_right + margin_up + height_left + diagonal
                                         Df Deviance   AIC
+ diagonal                   1   336.71 2016.7 <none>          336.71 2016.7
<none>                      1   338.37 2021.8 - diagonal      1   338.37 2021.8
- height_left                 1   342.03 2035.6 - margin_up    1   340.37 2030.5
- margin_up                   1   342.11 2035.9 - height_left  1   340.62 2031.5
- height_right                1   346.50 2054.6 - height_right 1   344.97 2050.1
- length                      1   460.40 2470.4 - length       1   455.03 2455.2

```

FIG. 2 – AIC

Le meilleur modèle est celui qui minimise l'AIC cet à dire celui de la forme :

$$\text{margin_low length} + \text{height_right} + \text{margin_up} + \text{height_left} + \text{diagonal}$$

- Étude du critère BIC

```

Start: AIC=2961.16          Step: AIC=2108.21          Step: AIC=2068.24          Step: AIC=2056.74
margin_low ~ 1               margin_low ~ length        margin_low ~ length + height_right    margin_low ~ length + height_right + margin_up
                               Df Deviance   AIC           Df Deviance   AIC           Df Deviance   AIC           Df Deviance   AIC
+ length                     1   357.83 2108.2 + height_right  1   346.45 2068.2 + margin_up     1   342.03 2056.7 + height_left  1   338.37 2048.3
+ margin_up                  1   524.22 2666.9 + margin_up     1   351.57 2089.7 + height_left  1   342.11 2057.1 <none>          342.03 2056.7
+ height_right                1   545.69 2725.6 + height_left  1   351.64 2090.0 <none>          346.45 2068.2 + diagonal    1   340.62 2058.0
+ height_left                 1   585.22 2827.9 <none>          357.83 2108.2 + diagonal    1   344.99 2069.3 - margin_up    1   346.45 2068.2
+ diagonal                   1   636.21 2950.1 + diagonal    1   356.55 2110.3 - height_right 1   357.83 2108.2 - height_right 1   351.57 2089.7
<none>                      1   644.23 2961.2 - length       1   644.23 2961.2 - length       1   545.69 2725.6 - length       1   476.64 2535.0
                                         Step: AIC=2048.28
                                         margin_low ~ length + height_right + margin_up + height_left
                                         Df Deviance   AIC
<none>                      338.37 2048.3
+ diagonal                   1   336.71 2048.4
- height_left                 1   342.03 2056.7
- margin_up                   1   342.11 2057.1
- height_right                1   346.50 2075.7
- length                      1   460.40 2491.5

```

FIG. 3 – BIC

Le meilleur modèle est celui qui minimise l'AIC cet à dire celui de la forme :

$$\text{margin_low length} + \text{height_right} + \text{margin_up} + \text{height_left} + \text{diagonal}$$

Après analyse, les deux méthodes nous permettent de conclure que nous devons garder l'ensemble des variables dans l'imputation.

Ps : nous n'avons pas utilisé la variable target pour l'imputation afin d'éviter d'influencer les futures prédictions.



- Imputation

Nous avons donc déterminer la présence de valeurs manquantes ainsi que leur localisation dans le dataset, ensuite nous pris la décision de les remplacer en utilisant la méthode GLM et pour cela, nous avons étudié les critères AIC et BIC afin de déterminer les meilleures variables à utiliser. Ainsi, nous pouvons alors remplacement des valeurs manquantes dans notre dataset afin de continuer l'étude sereinement et d'employer l'ensemble des méthodes souhaité sans nous soucier des valeurs manquantes.



2.2 Visualisation du jeux de données

2.2.1 Utilisation de Pairplot

Tout d'abord, nous voulons visualiser les données afin de mieux comprendre notamment leurs distributions. Ainsi, nous avons utilisé la représentation sous forme de *Pairplot*.

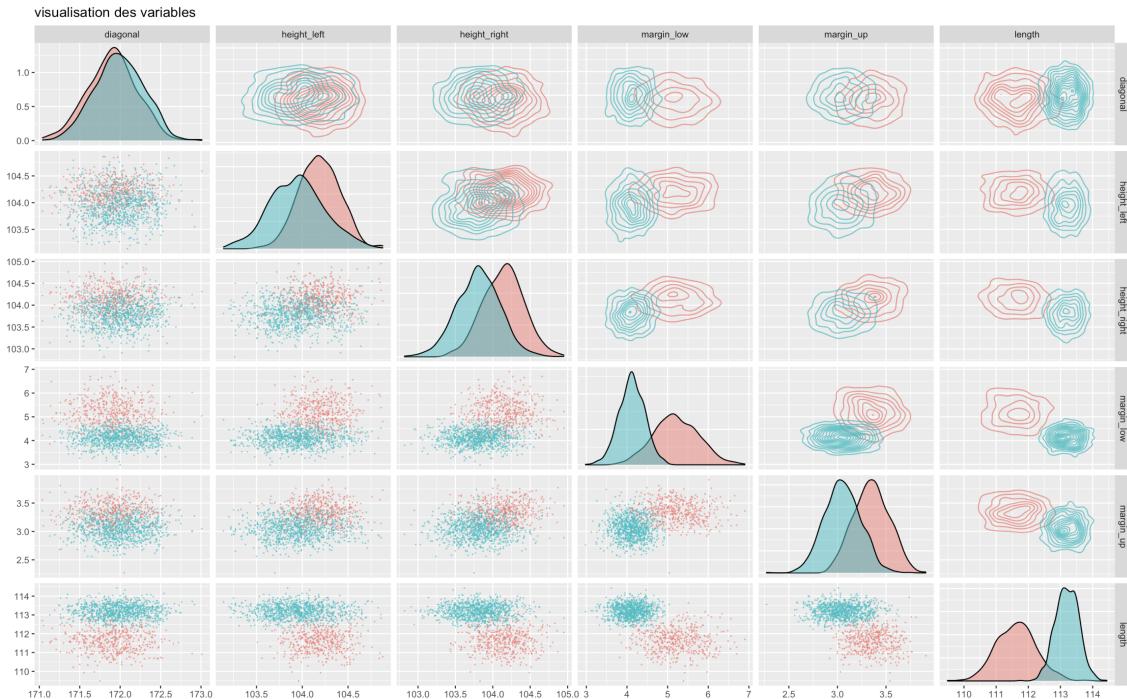


FIG. 4 – *Pairplot*

Nous avons représenté l'ensemble des variables en fonction d'elle-même, en faisant la distinction de couleur sur l'authenticité des billets.

Nous obtenons alors une matrice de graphes, elle est composée de trois parties distinctes avec trois représentations différentes de densités conditionnellement à leur authenticité (rouge = FAUX, bleu = VRAI) :

- le triangle supérieur : sous forme de courbes de niveaux des variables par rapport aux autres.
- la diagonale : sous forme de densité des variables par rapport à elles-mêmes.
- le triangle inférieur : sous forme de nuages de points des variables par rapport aux autres.

Partie diagonale de la matrice :

Mise à part la variable *diagonal*, où les densités des vrais et des faux billets sont confondus, toutes autres densités pour les autres variables ont des densités d'espérance bien distinctes en fonction de si ce sont de vrais ou de faux billets.

Nous remarquons également une nette séparation distinguant facilement les vrais et les faux billets entre la variable *length* et l'ensemble des autres variables. Ainsi, des modèles à séparation linéaire pourraient donner de bon résultat.



2.2.2 Utilisation de Boxplot

Visualisons maintenant les données sous une autre forme. Nous utilisons ici la représentation sous forme de *boxplot*, ainsi il sera aisément de déterminer la présence d'outlier ainsi que la variance et l'espérance de chaque variables en fonction de l'authenticité des billets.

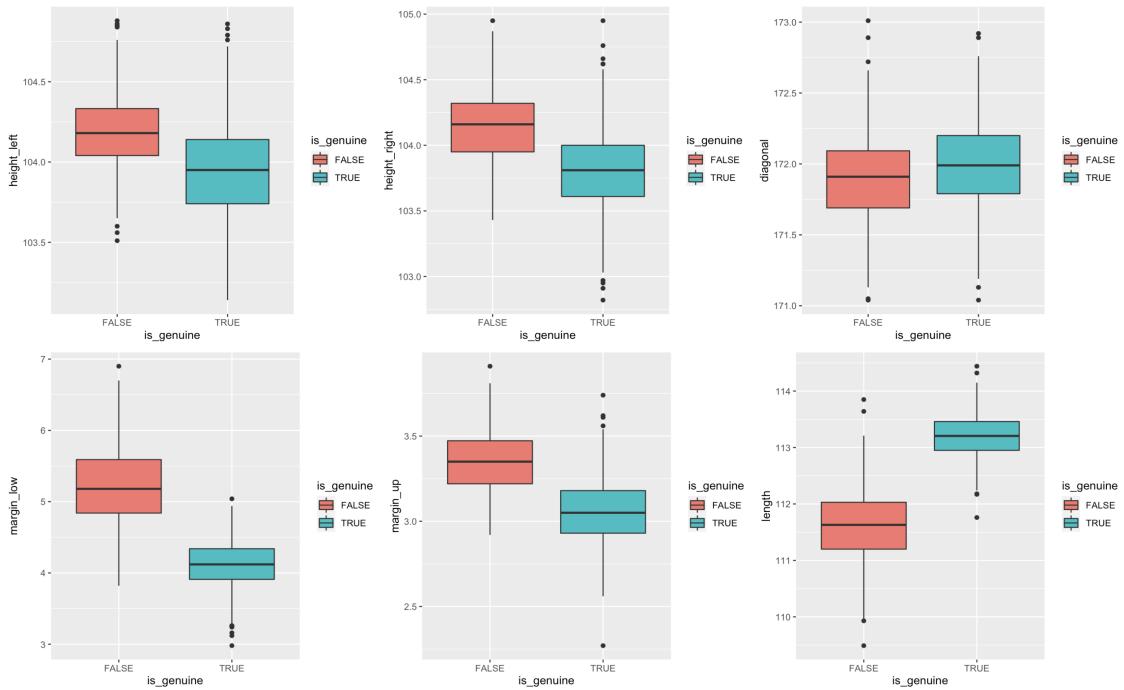


FIG. 5 – *Boxplot*

Nous remarquons à nouveau qu'à l'exception de la variable *diagonal*, toutes les autres variables font aisément le distinguo en fonction de si c'est de vrai ou de faux billets grâce notamment à leurs variances souvent très différents.

Nous avons donc vérifier le fait que *margin_low*, *length* et *height_left* ont des variances différentes avec un test de bartlett combiné à un modèle d'ANOVA.

```
Bartlett test of homogeneity of variances
Bartlett's K-squared = 216.89, df = 1, p-value < 2.2e-16
Bartlett test of homogeneity of variances
Bartlett's K-squared = 53.276, df = 1, p-value = 2.898e-13
Bartlett test of homogeneity of variances
Bartlett's K-squared = 207.23, df = 1, p-value < 2.2e-16
```

FIG. 6 – *BARTLETT*

Après analyse nous pouvons rejeter l'hypothèse d'homogénéité des variances.

De plus, nous observons la présence d'outliers sur l'ensemble des variables guidant ainsi le choix des modèles employés.

Les modèles paramétriques y sont sensibles, ainsi dans l'éventualité où nous utiliserons ce type de modèle, il faudra envisager de standardiser nos variables où d'appliquer *log* dessus.

Les modèles non-paramétriques tels que les arbres, random forest ou KNN sont quand à eux beaucoup plus



résilients face à ce type de situations, aucune modification ne sera donc nécessaire.



2.3 Rééchantillonnage

Notre dataset étant composée de deux tiers de FAUX et d'un tiers de VRAI il y a un léger déséquilibre dans la représentation des variables ainsi, nous pouvons effectuer un rééchantillonnage afin de minimiser cet écart entre les deux représentations.

Ps : le rééchantillonnage n'était pas impératif au vu du faible déséquilibre des classes.

Étant donné le faible nombre d'observation, nous allons plutôt opter pour une méthode d'oversampling pour remédier au problème. Pour cela, nous allons faire du bootstrapping, cependant il faudra faire attention, car cette méthode favorise le sur-ajustement.

Dans la suite de l'étude, nous allons donc utiliser ce nouveau dataset, composé de classes plus équilibré à 50% de VRAI et à 50% de FAUX.



3 Modèles de classifications

Après encodage de la variable cible, nous obtenons :

- 0 = False
- 1 = True

Nous allons chercher à minimiser les erreurs sur les faux positif cet à dire, diminuer le risque de dire qu'un faux billet est un vrai. Nous allons donc utiliser la métrique recall.

Nous utiliserons également la métrique précision pour évaluer l'erreur faite en affirmant qu'un billet est faux alors qu'il est vrai.

Et enfin, on utilisera la métrique F1 qui combine les deux.

Nous avons représenté les différentes métriques d'évaluation pour les différentes méthodes.

3.1 Régression logistique

3.1.1 Métriques d'évaluation

modele	accuracy	f1	recall	precision
	<dbl>	<dbl>	<dbl>	<dbl>
regression logistique	0.9973333	1	1	0.996
1 row				

FIG. 7 – Regression logistique métriques

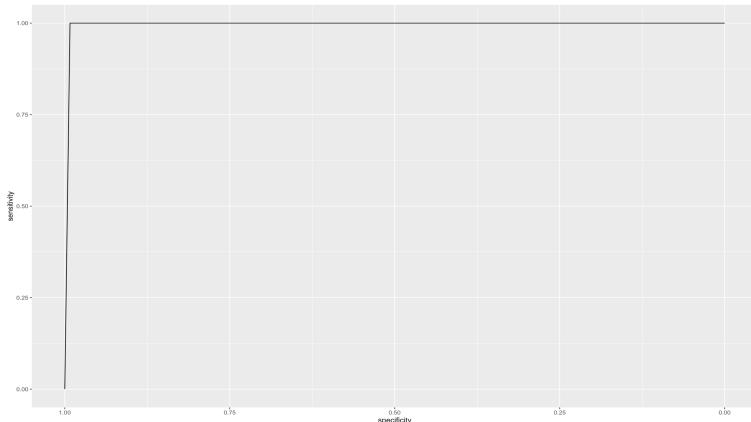


FIG. 8 – Regression logistique ROC

Ainsi nous calculons l'air sous la courbe afin d'avoir des estimations plus précises.

AUC = 0.996

L'air sous la courbe est très proche de 1 ainsi la regression logistique est un bon modèle. Les erreurs de ce modèle viennent de *accuracy* et de la *precision*.



3.2 Arbre de classification

La table de contingence nous montre de très bon résultat pour l'arbre.

3.2.1 Métriques d'évaluation

modele <chr>	accuracy <dbl>	f1 <dbl>	recall <dbl>	precision <dbl>
tree	0.9866667	1	0.9959839	0.984127
1 row				

FIG. 9 – Arbre de classification métriques

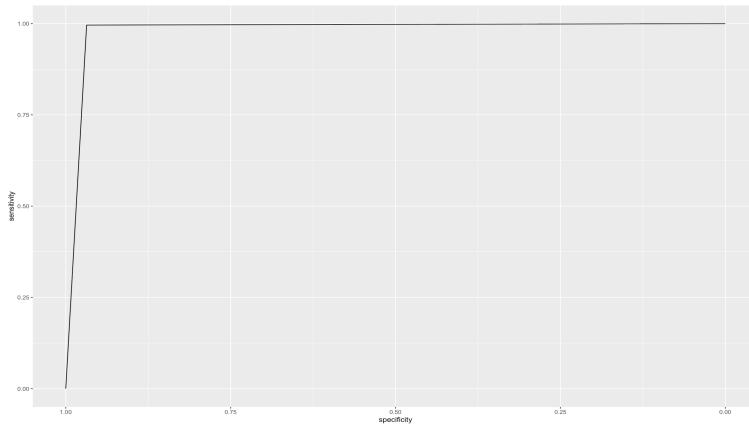


FIG. 10 – Arbre de classification ROC

Ainsi nous calculons l'air sous la courbe afin d'avoir des estimations plus précises.

AUC = 0.982

L'air sous la courbe est très également proche de 1 ainsi l'arbres de classification est un bon modèle. Les erreurs de ce modèle viennent de *accuracy* de *recall* et de la *precision*.



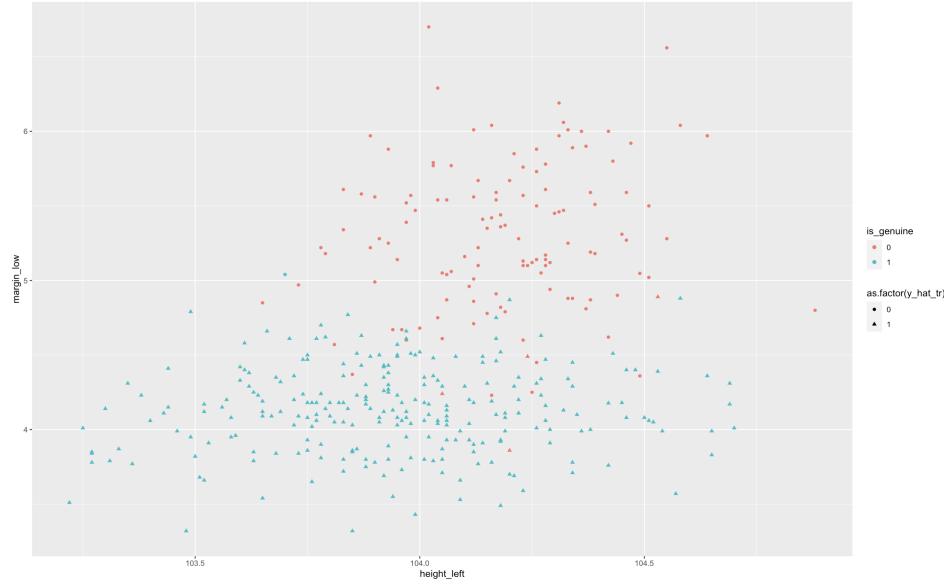


FIG. 11 – Résultat de la prédiction

Grâce à ce graphique nous remarquons une très bonne prédiction de la part de notre modèle mise à part quelques minimes erreurs.

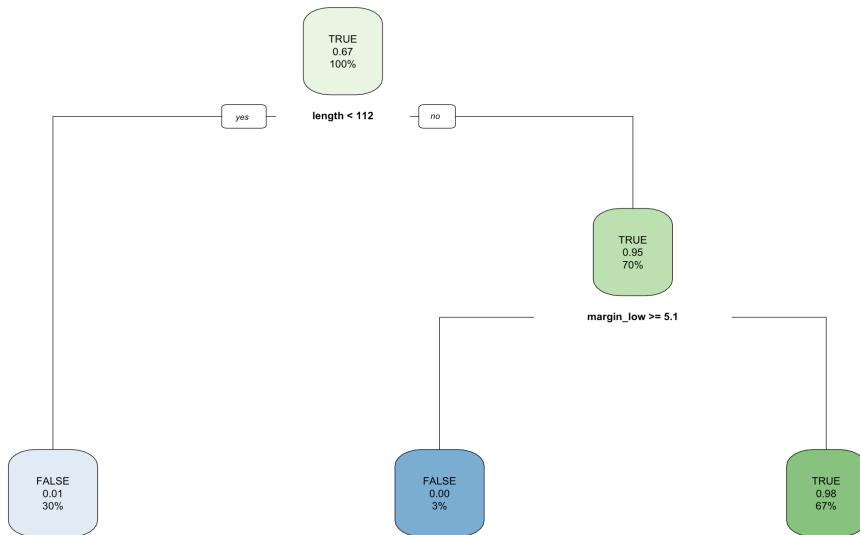


FIG. 12 – Arbre de classification

Sur cette figure nous avons représenté le fonctionnement de notre arbre de classification.



3.3 SVM

3.3.1 Hyperparamètres / Cross-validation

Nous avons réalisé une GridSearch combinée à une cross-validation de type 5-Fold afin de trouver le meilleur kernel.

Il est normal d'avoir une meilleure performance avec le kernel linéaire, car nous avons vu plus haut que la variable cible avait une belle séparation linéaire pour certaines variables.

Si nous avions choisi un autre kernel, nous aurions eu des résultats nettement moins bons. Ci-dessous, on peut voir la courbe pour le SVM avec un kernel sigmoid.

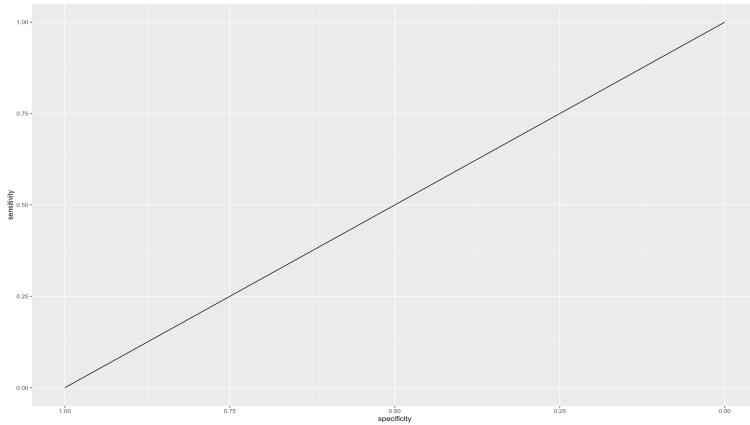


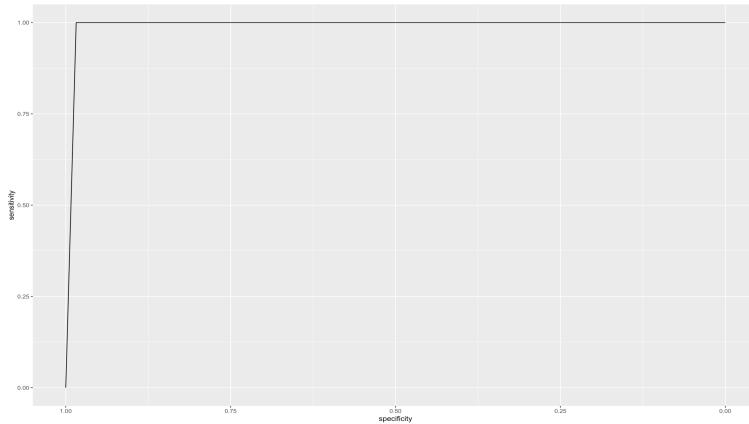
FIG. 13 – SVM ROC kernel sigmoid

Cette courbe montre très clairement que le kernel sigmoid n'est pas du tout adapté. Les labels sont attribué aléatoirement



3.3.2 Métriques d'évaluation

modele	accuracy	f1	recall	precision
<chr>	<dbl>	<dbl>	<dbl>	<dbl>
svm	0.9946667	1	1	0.9920319
1 row				

FIG. 14 – *SVM métriques*FIG. 15 – *SVM ROC kernel linéaire*

Ainsi nous calculons l'air sous la courbe afin d'avoir des estimations plus précises.
AUC = 0.992

L'air sous la courbe est très également proche de 1 ainsi, l'arbres de classification est un bon modèle. Les erreurs de ce modèle viennent de *accuracy* et de la *precision*.



3.4 ACP

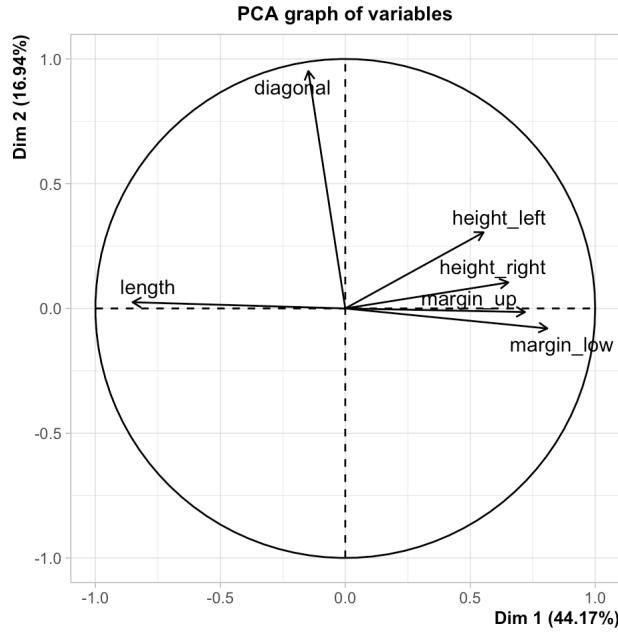


FIG. 16 – ACP

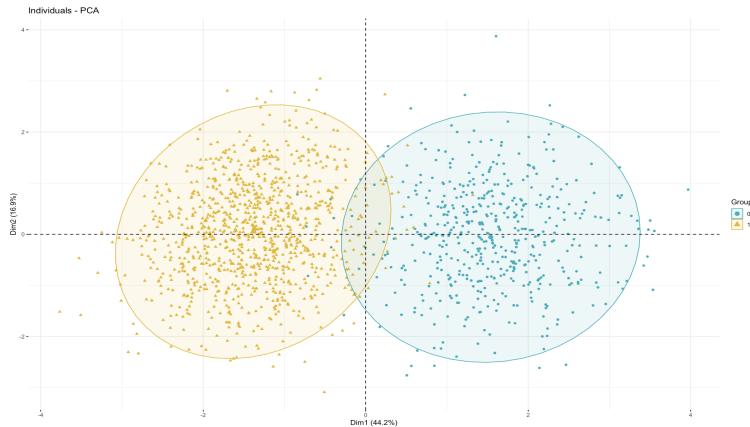


FIG. 17 – ACP

Nous remarquons que la deuxième composante (verticale) de l'ACP explique très peu la variable *is_genuine*. De plus, cette composante est fortement corrélée à la variable *diagonal* qui est très faiblement impacté par la variable target. Ceci coïncide bien avec le début de notre étude où nous disions que la variable *diagonal* explique mal les deux classes.

Nous remarquons également que la première composante (horizontale) explique bien la variabre cible. La classe 1 lui étant anti-corrélé tandis que la classe 0 lui est corrélé.

length est anti-corrélé à la premeire composante, tandis que *height_left*, *height_right*, *margin_up* et *margin_low* lui sont corrélé.

Nous constatons également que l'ACP fais des erreurs de prédictions, cette méthode est moins précise que toutes les autres.



3.5 Forêts aléatoires

3.5.1 Hyperparamètres / Cross-validation

Nous avons effectué une GridSearch sur les hyperparamètres mtry (nombres de variables utilisé à chaque split) et ntree (nombres d'arbres).

Nous avons pris ces différents valeurs pour ces différents hyperparamètres :

- $mtry = \{1, 2, 3\}$
- $ntree = \{10, 50, 100\}$

Et à chaque combinaisons d'hyperparamètres nous avons effectué une cross validation du type 5-Fold.

Nous exhibons ici le meilleur résultats :

```
"model: RandomForest  mtry:1  ntree: 50  precision: 0.993243243243243  recall:0.986577181208054"
"model: RandomForest  mtry:1  ntree: 50  precision: 0.986111111111111  recall:0.986111111111111"
"model: RandomForest  mtry:1  ntree: 50  precision: 0.974025974025974  recall:1"
"model: RandomForest  mtry:1  ntree: 50  precision: 0.993377483443709  recall:0.993377483443709"
"model: RandomForest  mtry:1  ntree: 50  precision: 1  recall:1"
```

FIG. 18 – Random forest GridSearch

3.5.2 Métriques d'évaluation

modele	accuracy	f1	recall	precision
<chr>	<dbl>	<dbl>	<dbl>	<dbl>
random Forest	0.992	1	0.9959839	0.992
1 row				

FIG. 19 – Random forest métriques

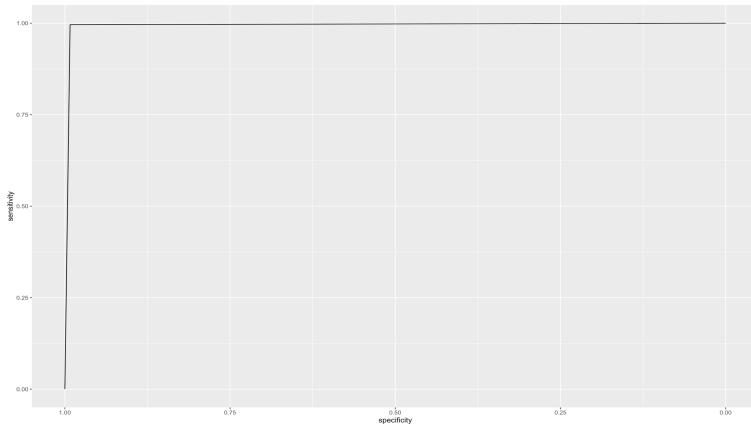


FIG. 20 – Random forest ROC

Ainsi nous calculons l'air sous la courbe afin d'avoir des estimations plus précises.
AUC = 0.996

L'air sous la courbe est très également proche de 1 ainsi random forest est un bon modèle. Les erreurs de ce modèle viennent de *accuracy* de *recall* et de la *precision*.



3.6 KNN

3.6.1 Hyperparamètres / Cross-validation

Nous cherchons cette fois-ci à optimiser l'hyperparamètre k , pour cela nous employons la méthode du coude.

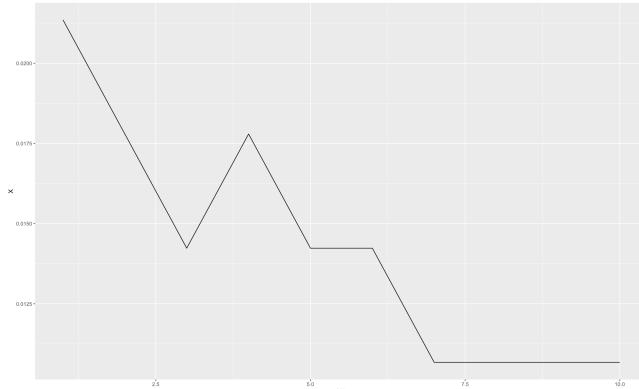


FIG. 21 – KNN elbow-method

Il semblerait que la meilleure valeur k se trouve entre 6 et 7. Nous confirmons effectivement en effectuant une GridSearch avec des valeurs de k allant de 1 à 10.

3.6.2 Métriques d'évaluation

modele	accuracy	f1	recall	precision
<chr>	<dbl>	<dbl>	<dbl>	<dbl>
knn				
1 row	0.9946667	1	1	0.9920319

FIG. 22 – KNN métriques

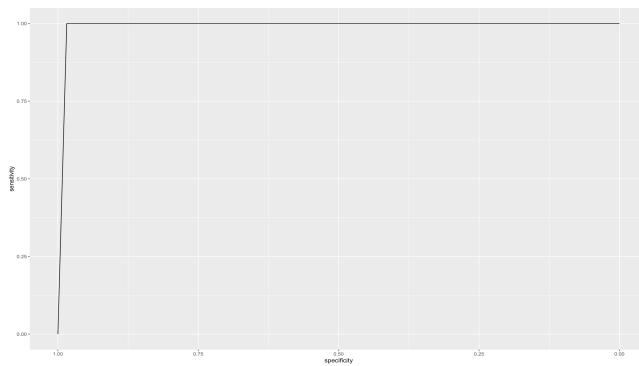


FIG. 23 – KNN ROC

Ainsi, nous calculons l'aire sous la courbe afin d'avoir des estimations plus précises.
AUC = 0.99

L'air sous la courbe est très également proche de 1 ainsi, l'arbre de classification est un bon modèle. Les erreurs de ce modèle viennent de *accuracy* de *recall* et de la *precision*.



4 Conclusion

Pour conclure, au fur et à mesure de l'étude nous avons observé de très bons résultats sur l'ensemble des méthodes. Ceci est en partie expliqué par une bonne dissociation des classes.

Les deux meilleurs modèle en terme d'AUC sont le random forest et la regression logistique. Cependant le random forest fait des erreurs supplémentaires au niveau du *recall* ce qui caractérise une plus grande erreur sur la prédiction de faux billets, il prédit à tort que certains billets sont vrais alors qu'ils sont faux.

Or, il s'agit de l'erreur la plus grave que l'on peut commettre dans le cas de notre étude, ainsi, il faut minimiser cette erreur au maximum et donc privilégier la régression logistique qui a un recall plus élevé (=1).

Et enfin le random forest est une méthode de boîte noire, difficilement interprétable contrairement à la régression logistique.

