VINCENT GAGNON – GAGV25059800
Baccalauréat en Informatique

Rapport TP1

Travail d'apprentissage machine
présenté à

Julien Maitre
dans le cadre du cours
8INF436 Forage de données

Module d'informatique et de mathématique
Université du Québec à Chicoutimi
Le 19 février 2021

Table des matières

Description de l'ensemble de données

L'ensemble de données contient des données sur 1066 observations concernant les éruptions solaires. Chaque instance possède 11 attributs et représente les caractéristiques capturées pour une région active du soleil. Il possède deux classes, soit il y a une éruption solaire dans les prochaines 24h ou non. Voici la description des caractéristiques.

largest_spot_size

Code représentant la tache la plus grosse, prend les valeurs X,R,S,A,H,K.

spot_distribution

Code pour la distribution de la tache, prend les valeurs X,O,I,C.

Activity

Activité de la zone, 1 = réduit et 2 = aucun changement.

Evolution

Évolution de la zone, 1 = désintégration, 2 = aucun changement et 3 = augmentation.

Previous_24_hour_flare_activity_code

Code d'activité de la zone dans les dernière 24h, 1 = plus petit que M1, 2 = 1 M1, 3 = plus que M1.

Historically_complex

Indique si la région est historiquement complexe, 1 = oui, 0 = non.

Did_region_become_historically_complex

Indique si la région est devenue historiquement complexe dans ce passage au travers du disque du soleil, 1 = oui, 2 = non.

Area

Aire de la région, 1 = petite, 2 = grande.

C-class_flares_production_by_this_regions

Nombre d'éruption solaire de classe C produites dans les dernières 24h.

M-class_flares_production_by_this_regions

Nombre d'éruption solaire de classe M produites dans les dernières 24h.

X-class_flares_production_by_this_regions

Nombre d'éruption solaire de classe X produites dans les dernières 24h.

class

Éruption solaire a eu lieu dans les prochaines 24h, 1 = oui, 0 = non. Il y a 43 instances dans la classe oui et 1023 instances dans la classe non.

Vérification et pré-traitement des données

L'ensemble de données ne possède aucune valeur manquante et donc il n'était pas nécessaire d'utiliser un algorithme de remplacement de valeurs manquantes.

L'ensemble de données possède deux attributs catégoriel représenté par des lettres, il faut donc les transformer afin de pouvoir appliquer les différents algorithmes de réduction de dimensionnalité. On utilise donc le LabelEncoder de sklearn afin de transformer les classes des attributs en nombres entiers allant de 0 à n-1 nombre de classes.

Le reste des attributs qui ne sont pas transformés avec le LabelEncoder est normalisé afin de ne pas biaiser les algorithmes de réduction de dimensionnalité.

Finalement, l'attribut class qui représente la classe d'une instance est débalancé, on y retrouve 43 cas d'éruptions solaires contre 1023 cas dans lequel on ne retrouve aucune éruption. On utilisera donc la méthode de reconstruction de la réduction de dimensionnalité afin de détecter les instances anormales qui sont dans la classe des éruptions solaires.

Classification des instances du dataset de test

Maintenant que les données sont vérifiées et prêtes à l'utilisation, on appliquera différents algorithmes de réduction de dimensionnalité pour ensuite essayer de reconstruire les données d'origine avec l'ensemble d'attributs réduit. On utilisera des outils de visualisation des données afin de mieux voir les différences entre les différents algorithmes. On testera aussi différents ratios de division pour le dataset d'entrainement et de test.

Hypothèse

Selon moi, le nombre faible d'attributs dans l'ensemble de données, soit 11, ne permettra pas à la réduction de dimensionnalité et à la détection d'anomalie d'être efficace et la précision moyenne de la courbe précision-rappel du modèle ne sera pas en haut de 60%, peu importe le ratio de séparation des données ou l'algorithme utilisé. Je crois aussi qu'on ne sera pas en mesure de distinguer facilement les instances avec seulement les deux composants principaux dans une visualisation.

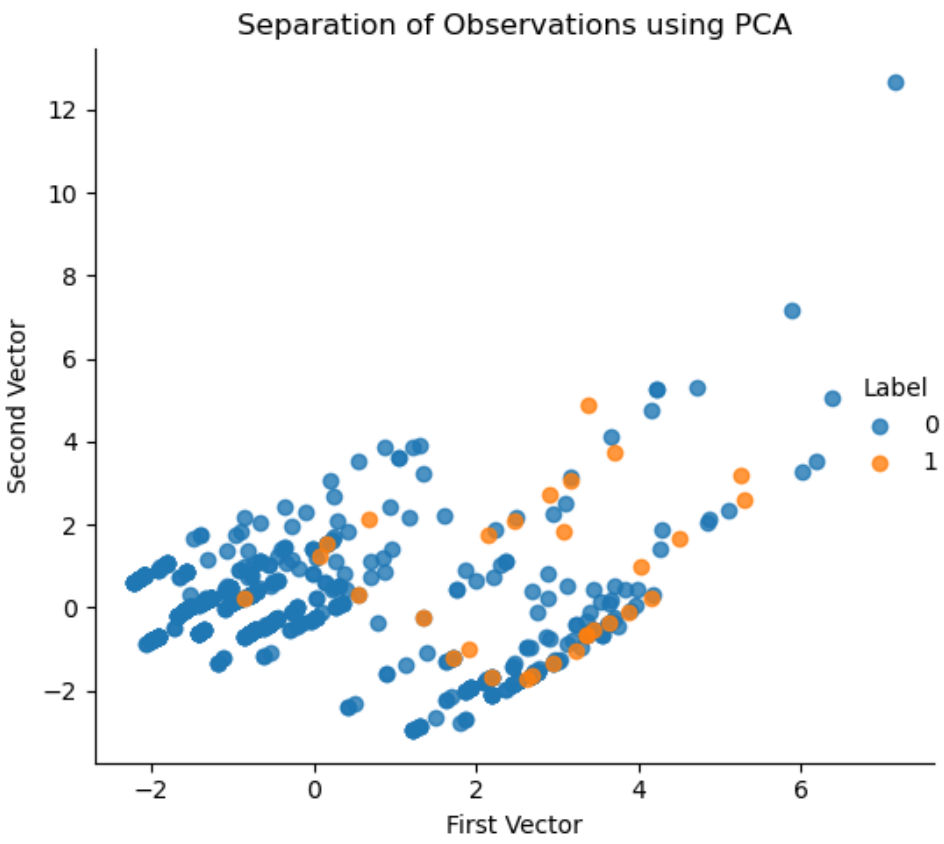
Méthodologie

Étant donné le nombre rare d'éruptions solaires comparativement au total du nombre d'instances, nous utiliserons la réduction de dimensionnalité pour ensuite reconstruire les données avec les données réduite. On va ensuite comparer la différence entre les données d'origine et les données reconstruites afin de repérer ceux qui ont la plus grande différence. Ces instances seront probablement ceux qui sont les éruptions solaires car celles-ci sont plus rares et vont différer des instances qui ne sont pas des éruptions solaires. On aura donc un score d'anomalie, qui sera l'erreur de reconstruction, qui nous permettra de détecter les anomalies dans les instances et de retrouver ceux qui sont le plus propice d'être des éruptions solaires.

Nos métriques d'évaluation seront la courbe précision-rappel, la précision moyenne, l'aire sous la courbe ROC et quelques visualisations.

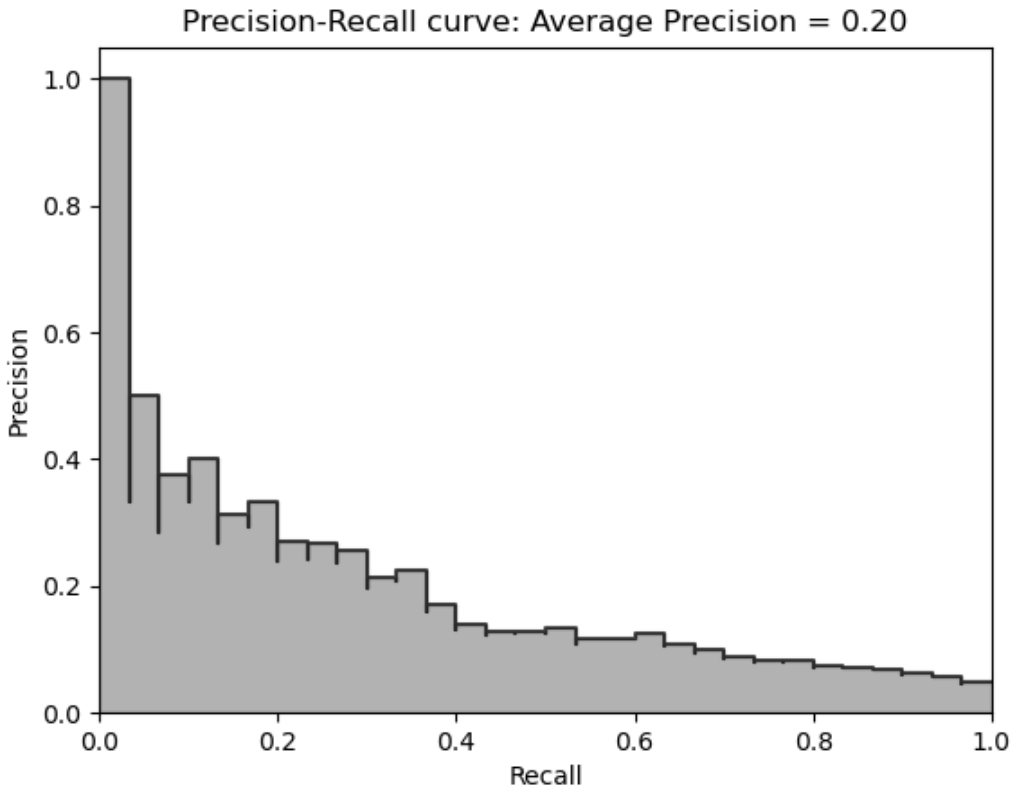
De plus, le score d'anomalie dépendra grandement du nombre de composants que l'on choisit de garder lors du PCA, on doit donc trouver le nombre de composants idéal à garder afin d'avoir la meilleure détection d'anomalie. Pour cela, on utilise l'algorithme PCA et on test différents nombre de composants principaux. Après plusieurs essais, le nombre de composants principaux qui donnait le meilleur résultat était 6, on a donc gardé ce nombre de composants principaux pour tester le reste des algorithmes. Voici les métriques d'évaluation avec 6 composants :

Premièrement, la séparation des instances en utilisant seulement les 2 premiers composants principaux. On voit que deux composants n'est clairement pas assez pour différencier les deux classes.



Deuxièmement, voici la courbe précision-rappel et la courbe ROC. On peut voir que le modèle PCA n'est pas très bon pour détecter les instances d'éruptions solaires dans ce dataset. Nous

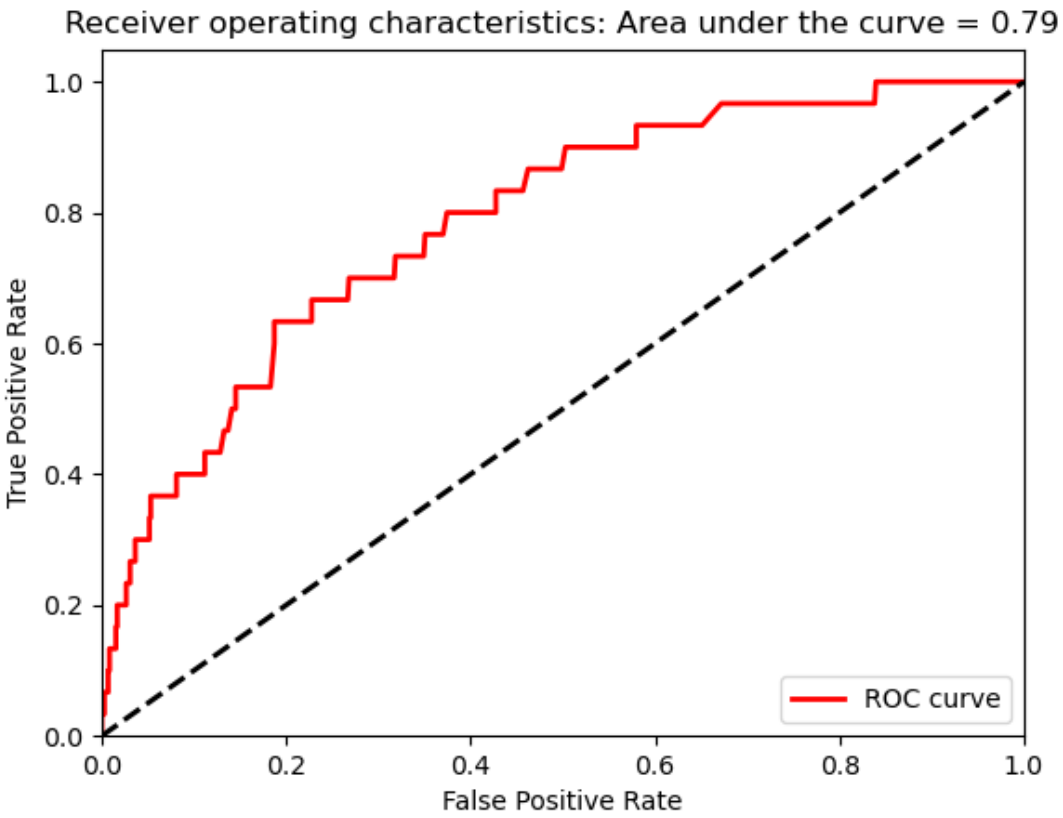
retrouvons un précision moyenne de 20%, ce qui est très mauvais.



Nous sommes en mesure de détecter 30% des éruptions solaires avec une précision de 21%. Au total c'est 9 éruptions sur 43 que PCA a pu trouver, ce qui est, encore une fois, pas super.

```
Precision:  0.21
Recall:  0.3
Solar flares Caught out of 43 Cases: 9
```

Finalement, la troisième visualisation nous permet de voir que nous avons une aire sous la courbe ROC de 0.79, nous devrons attendre de voir les résultats d'autres algorithmes de réduction de dimensionnalité avant d'interpréter correctement cette visualisation.

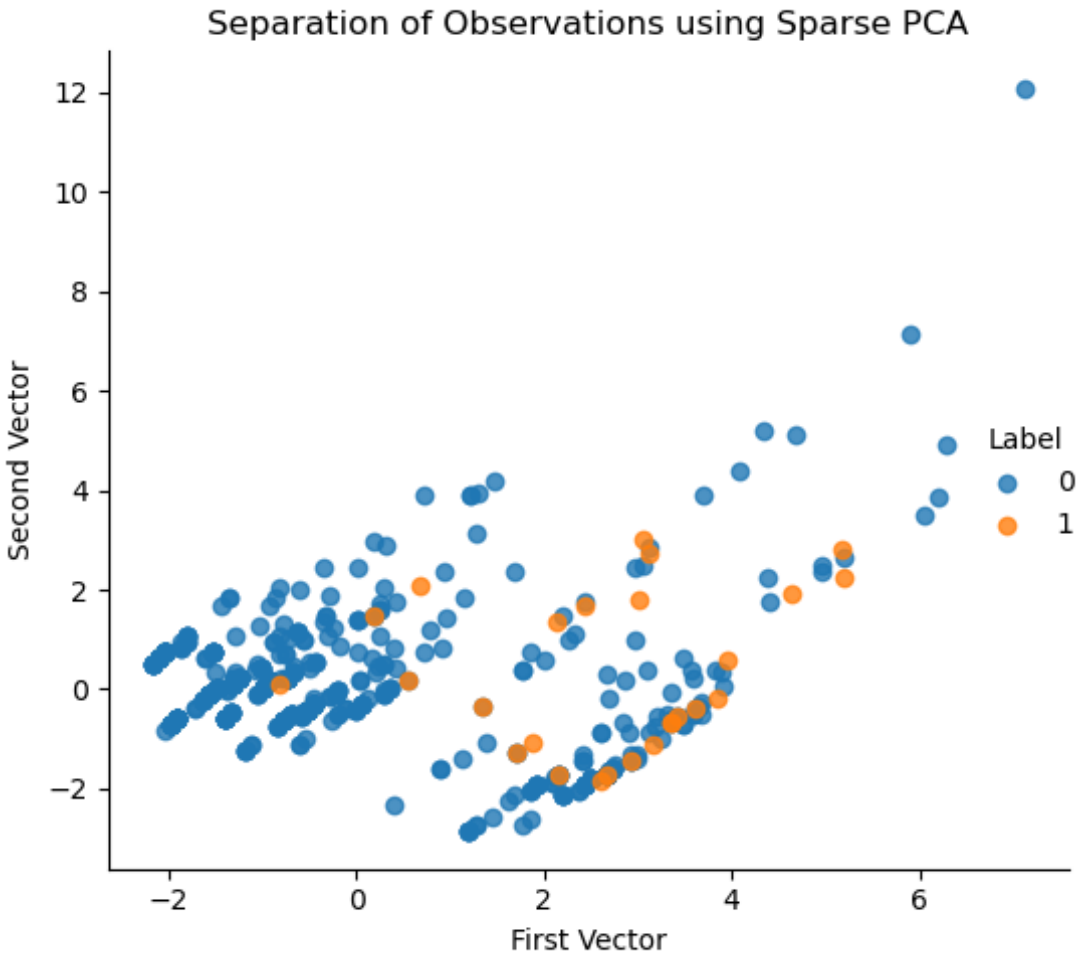


PCA n'étant clairement pas une bonne solution, nous allons essayer de détecter les éruptions solaires avec d'autres algorithme pour voir si on peut arriver à en trouver un qui arrive à faire un bon travail, en commençant par le Sparse PCA.

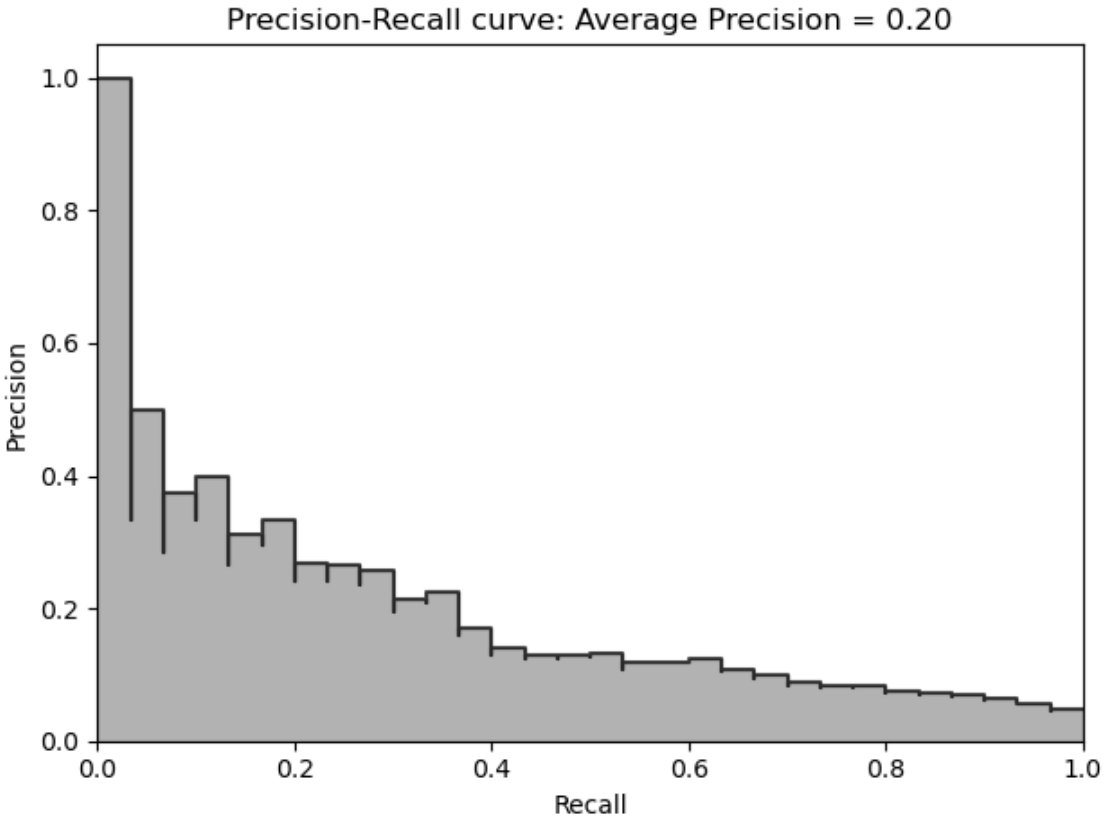
Sparse PCA

Le Sparse PCA étant similaire au PCA, on ne s'attend pas à voir une grande amélioration de la détection d'éruptions solaires. Sparse PCA est essentiellement une version moins dense de PCA en fournissant une représentation distribuée des principaux composants. En plus des paramètres de PCA, on doit également fournir un paramètre alpha, qui contrôle le degré de parcimonie. On pourra tester différentes valeurs afin de voir si cela peut améliorer la performance du modèle. On entraine donc notre modèle avec 6 composants comme nous l'avons déterminé au début de l'expérimentation et nous utiliserons un alpha de 0.0001 pour commencer.

Pour ce qui est de la séparation des instances avec seulement deux attributs, on peut voir que c'est très similaire à PCA, on n'arrive également pas à discerner les instances qui sont des éruptions solaires de ceux qui ne le sont pas.

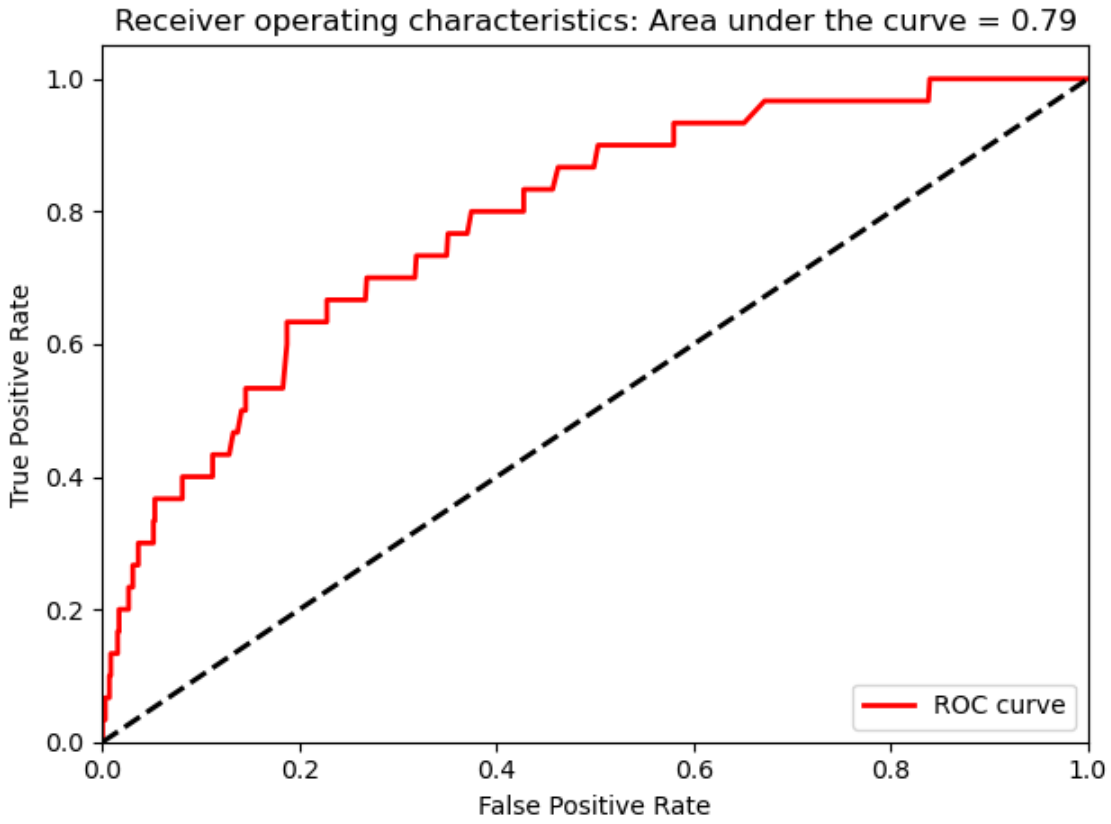


Ensuite, la courbe précision rappel nous démontre une précision moyenne de 20%, donc pareil à PCA, il en est de même pour le nombre d'éruptions détecté, la précision et le rappel.




```
Precision:  0.21
Recall:  0.3
Solar flares Caught out of 43 Cases: 9
```

Finalement, La courbe ROC est pareil que le PCA et il n'y a donc aucun avantage à utiliser Sparse PCA ou PCA.



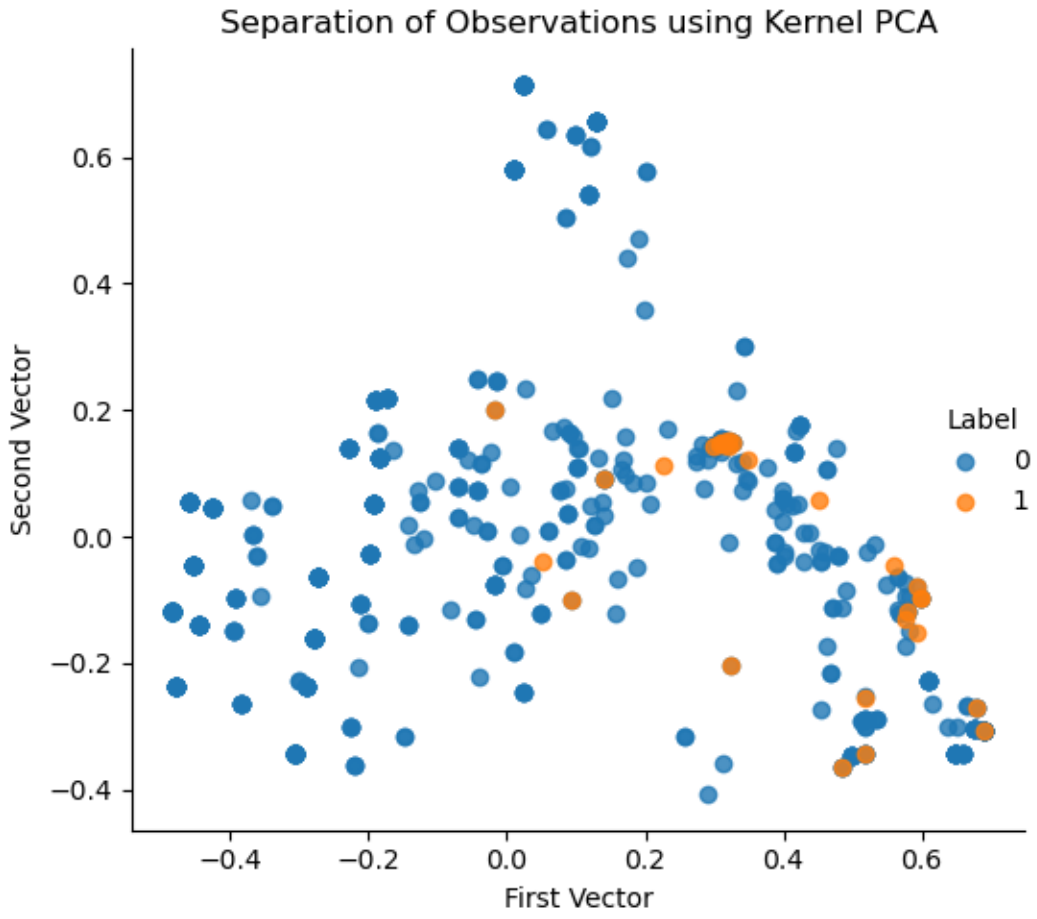
On a effectué des tests sur les ratios d'instances de test et d'entrainement et aussi sur la valeur alpha. Un ratio de 40% de valeurs de tests diminue la performance du modèle. Le changement de la valeur alpha quant à elle, n'a aucun impact sur la performance sauf lorsqu'elle est plus grande que 1, dans ce cas la performance diminue.

Cette solution étant très similaire à PCA et donc mauvaise, nous allons maintenant tester l'algorithme de réduction de dimensionnalité kernel PCA.

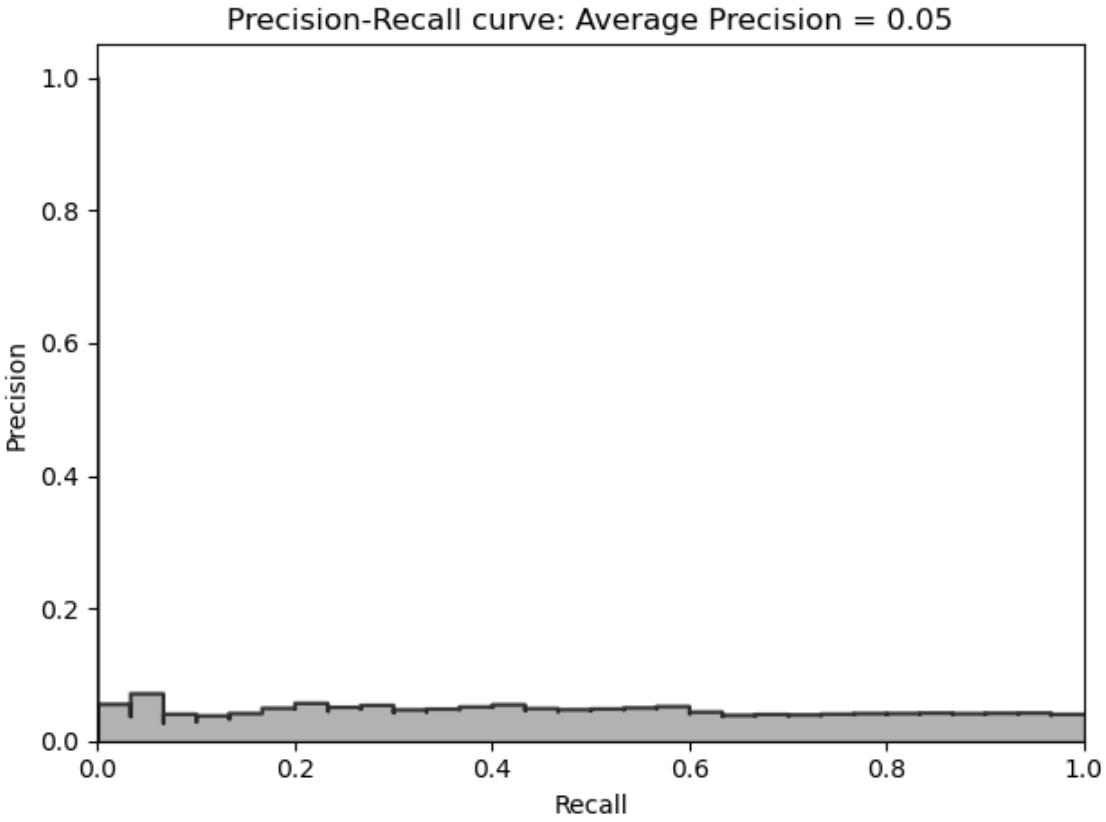
Kernel PCA

Kernel PCA est une forme non-linéaire de PCA qui est utile lorsque les instances d'éruptions solaires de sont pas linéairement séparable des instances normales. On utilisera encore 6 composants principaux.

Si l'on fait une visualisation des instances avec seulement les deux premiers composants principaux, on peut voir qu'il y a une légère distinction entre les instances qui sont des éruptions solaires et ceux qui ne le sont pas. En effet, les éruptions solaires ont l'air à avoir tendance à être plus vers le bas à droite du graphique. Cependant, ce n'est pas une énorme distinction.



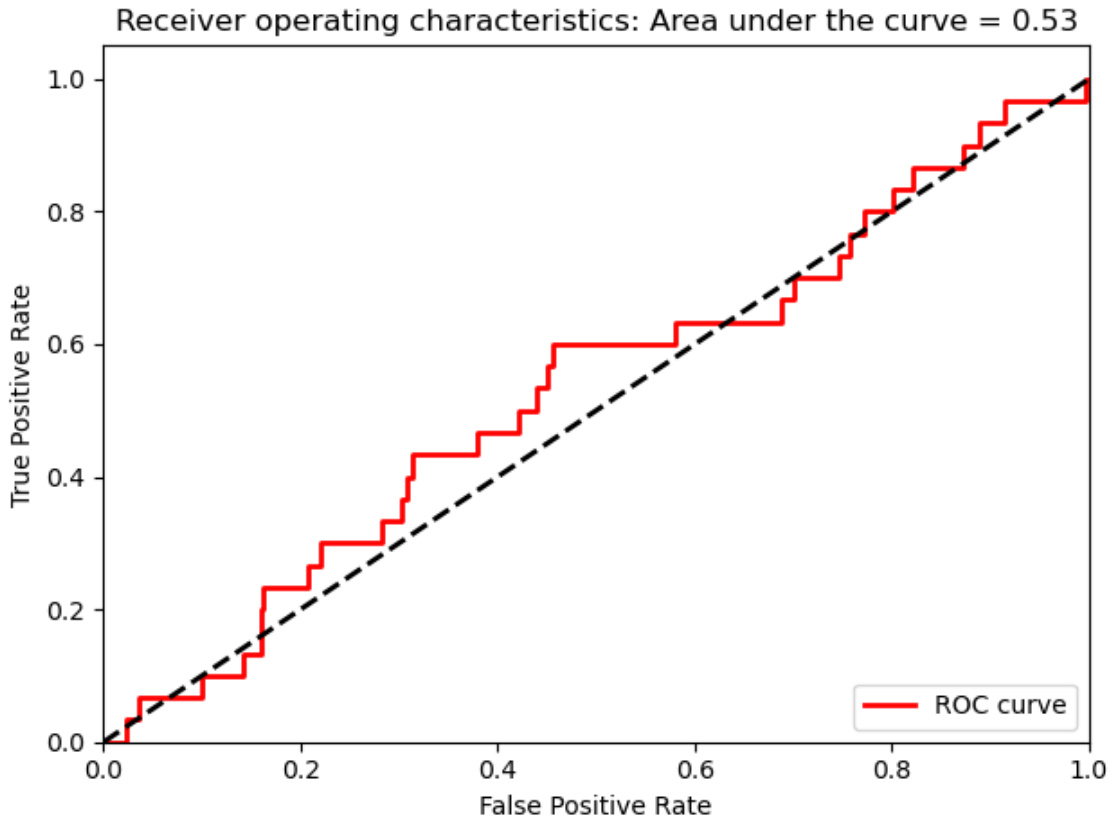
Ensuite nous avons la courbe de précision rappel :



Celle-ci nous démontre un précision moyenne piètre de 5%, avec 7% d'éruptions détectées et 5% de précision.

```
Precision:  0.05
Recall:  0.07
Solar flares Caught out of 43 Cases: 2
```

Finalement, l'aire sous la courbe ROC est encore une fois décevant. On n'y retrouve que 0.53 ce qui est bien pire que le PCA ou le Sparse PCA.

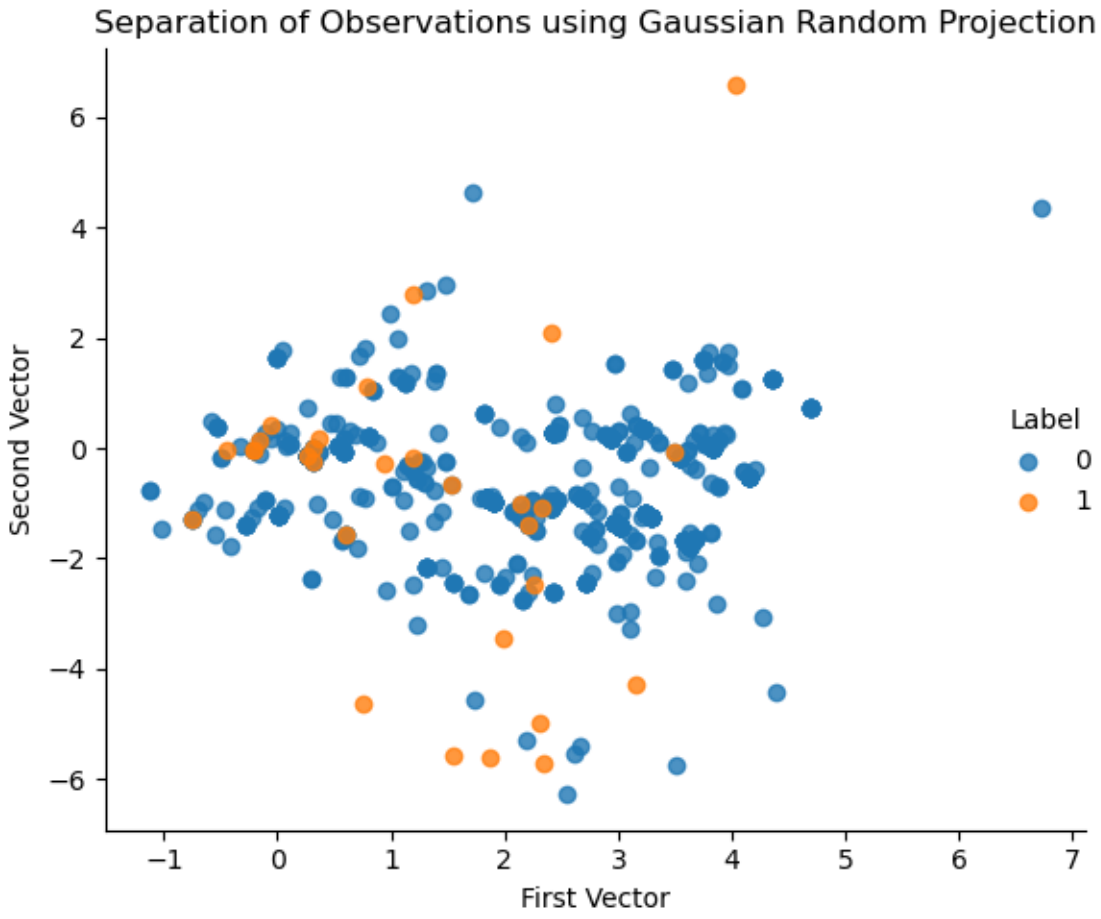


Ces métriques nous montrent que le kernel PCA n'est pas du tout adapté à cet ensemble de données. Même en changeant les paramètres de nombre de composant et le ratio de données de test les résultats restent mauvais. Cela s'explique probablement par le fait que l'ensemble de caractéristiques d'origine est linéairement séparable et qu'il n'est donc pas efficace d'utiliser le kernel PCA dans ce contexte.

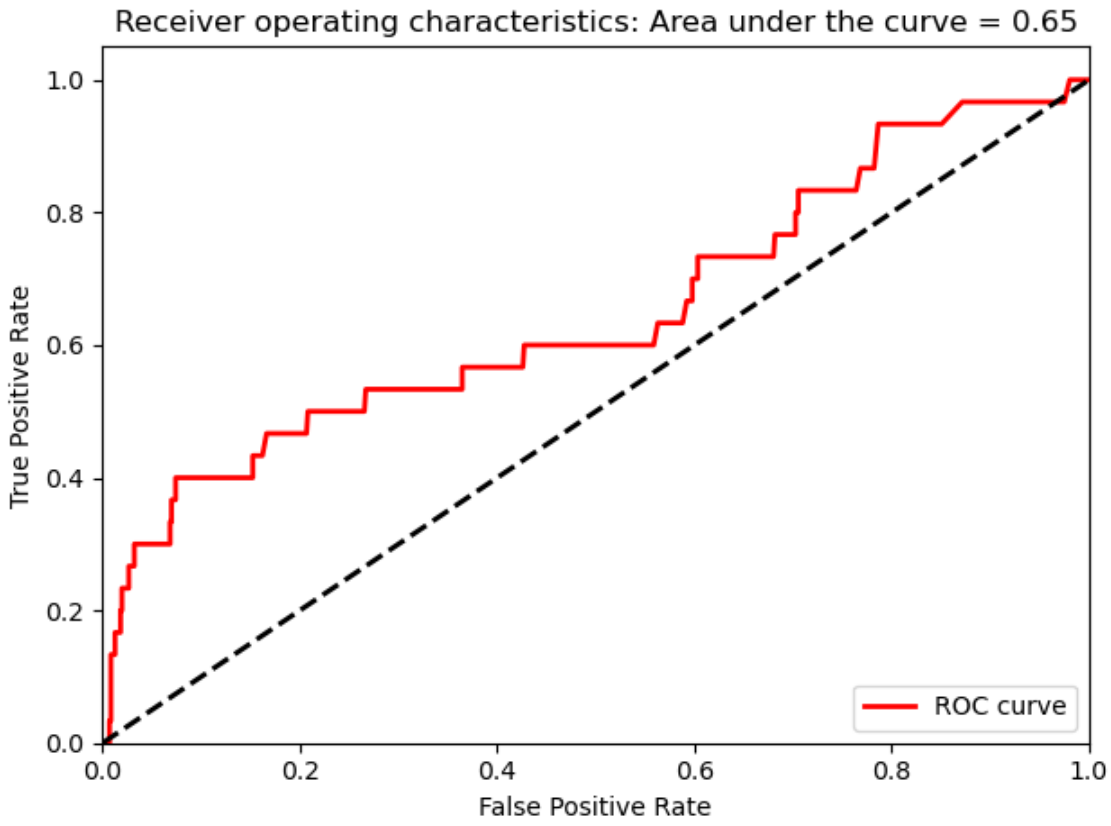
Gaussian Random Projection

Nous allons maintenant utiliser la projection aléatoire Gaussienne, elle possède le paramètre eps qui est différent des autres algorithmes, il contrôle la qualité de la réduction basée sur le lemme de Johnson-Lindenstrauss.

Une visualisation des deux premiers composants principaux ne nous aide aucunement à distinguer les instances d'éruptions solaires.



Pour les autres métriques nous retrouvons des résultats médiocres :



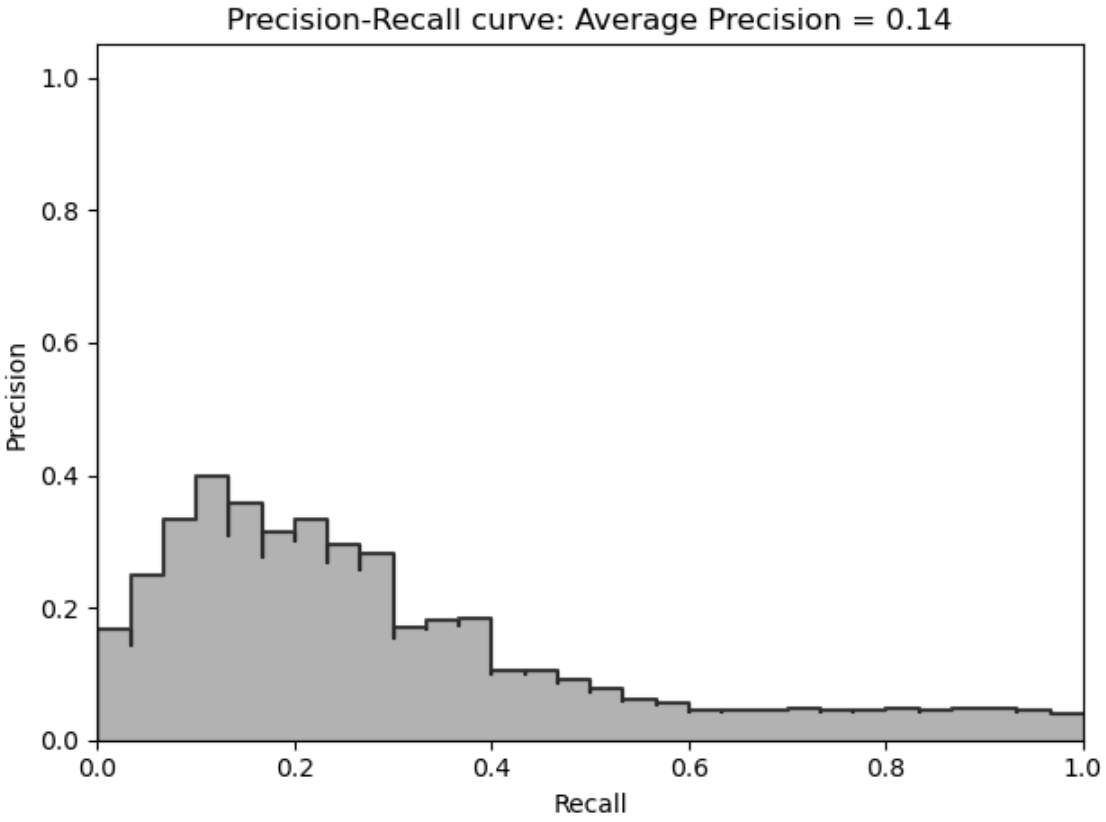
L'aire sous la courbe ROC est de seulement 0.65, ce qui est un peu mieux que kernel PCA mais très mauvais tout de même.

Pour ce qui est de la courbe précision-rappel, on obtient une précision moyenne de 14%, avec 30% d'éruptions solaires détectés et 21% de précisions, ce qui est pareil au PCA et Sparse PCA.

```
Precision:  0.21
Recall:  0.3
Solar flares Caught out of 43 Cases: 9
```

Cependant, passer de 6 composants principaux à 5 pour la reconstruction on augmente légèrement le nombre d'instances d'éruptions solaires découvertes, on passe de 9 à 11 et on obtient un pourcentage d'éruptions détecté de 37% et une précision de 26%, ce qui est meilleur que PCA. Cependant, le reste des métriques demeurent inchangés et pire que PCA.

```
Precision:  0.26
Recall:  0.37
Solar flares Caught out of 43 Cases: 11
```



On a aussi essayé de faire varier le paramètre eps afin de déterminer le nombre de composants à garder mais sans succès. Le nombre d'attributs dans l'ensemble de données original est trop petit pour appliquer le paramètre eps.

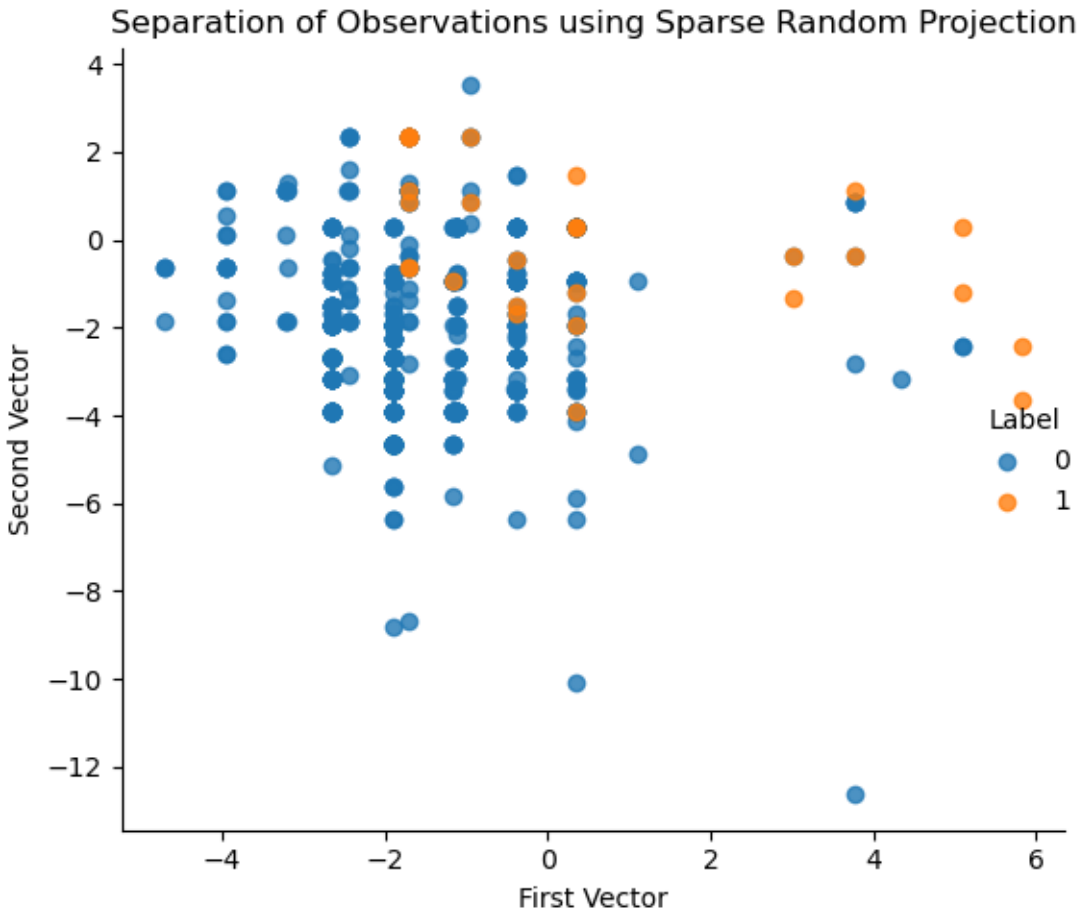
La projection aléatoire Gaussienne n'est donc pas le meilleur algorithme pour ce dataset.

Sparse Random Projection

Essayons maintenant le Sparse random projection. Celle-ci possède les mêmes paramètres que le Gaussian Random Projection en plus d'un paramètre dense_output que nous mettrons à True.

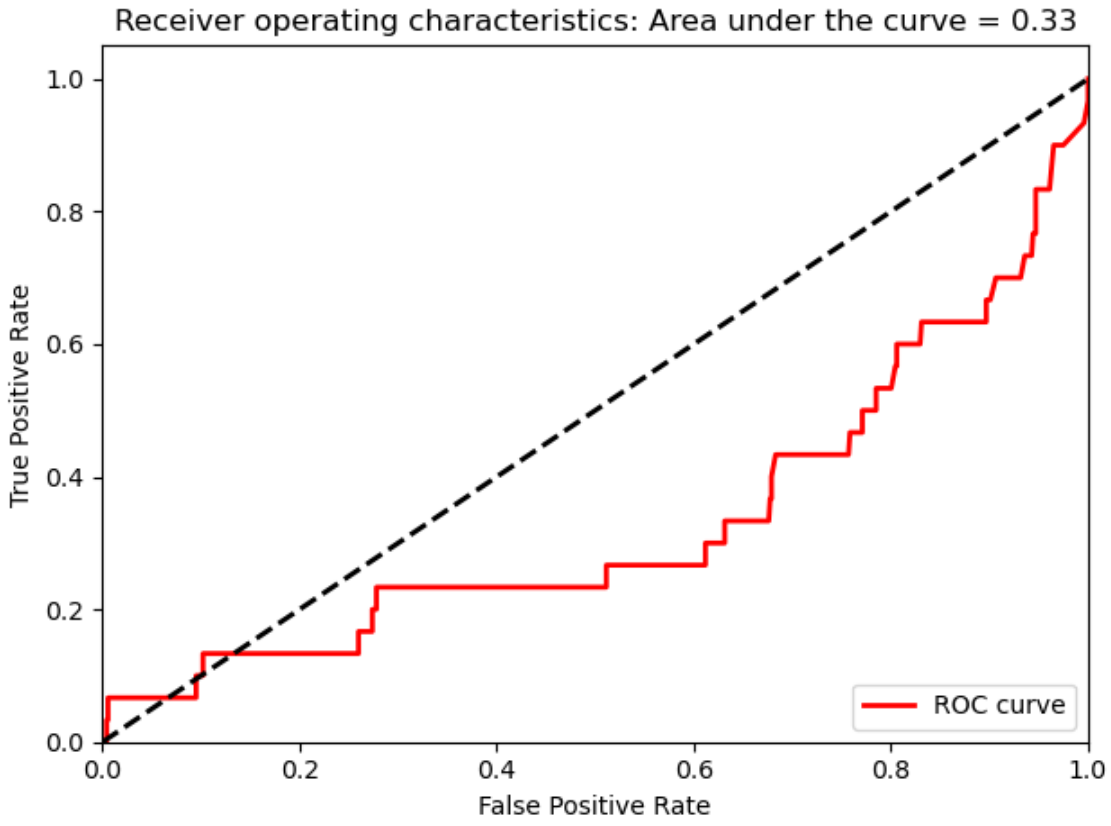
On utilise encore 6 composants principaux tel que défini plus tôt.

Voici la représentation des instances selon les deux composants principaux :

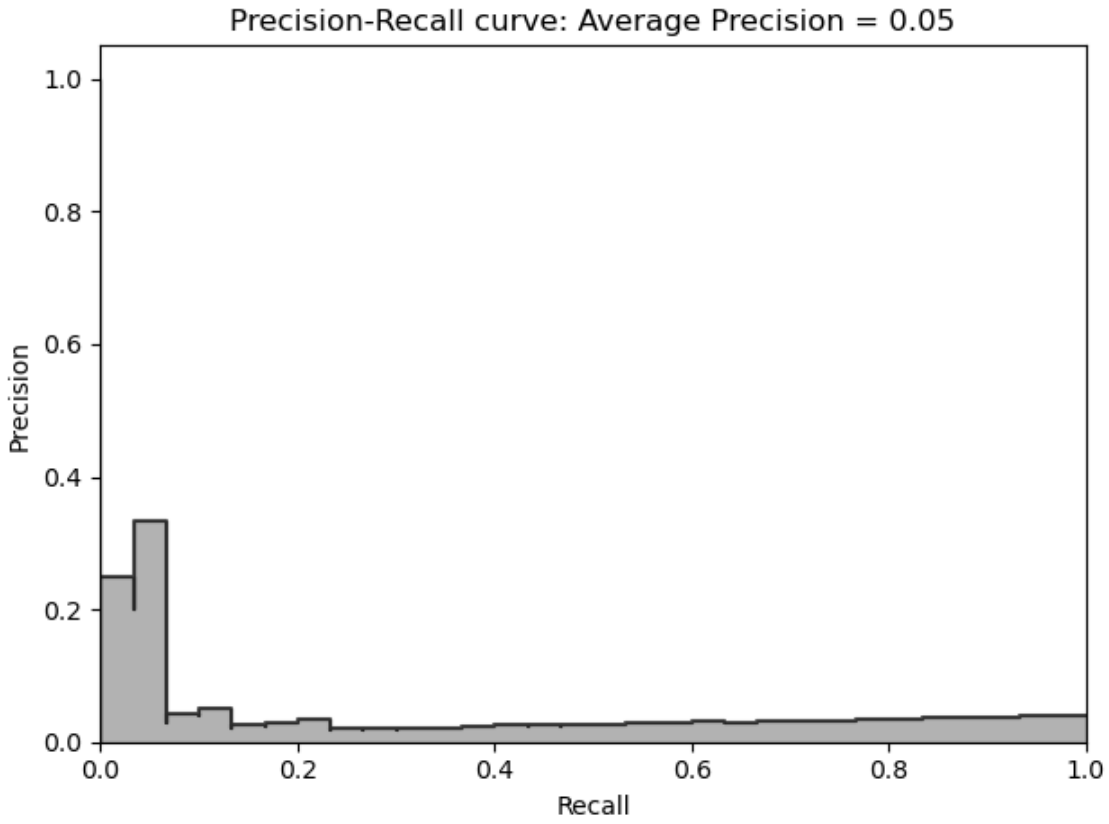


On peut voir que cela est très différent des autres algorithmes, les instances suivent les lignes des valeurs de axes. On peut remarquer que les instances d'éruptions solaires se rapprochent légèrement du haut droit du graphique, mais cela est une tendance très légère.

L’aire sous la courbe ROC est de 0.33, mais en inversant les attributs on aurait une aire de 0.66, ce qui est encore mauvais et très similaire à Gaussian Random Projection.



La courbe précision rappel elle ne montre rien de bon, avec une précision moyenne de 5%, donc 7% des instances d'éruptions solaires détectées et 5% de précision, c'est de loin le pire algorithme jusqu'à présent.




```
Precision:  0.05
Recall:  0.07
Solar flares Caught out of 43 Cases: 2
```

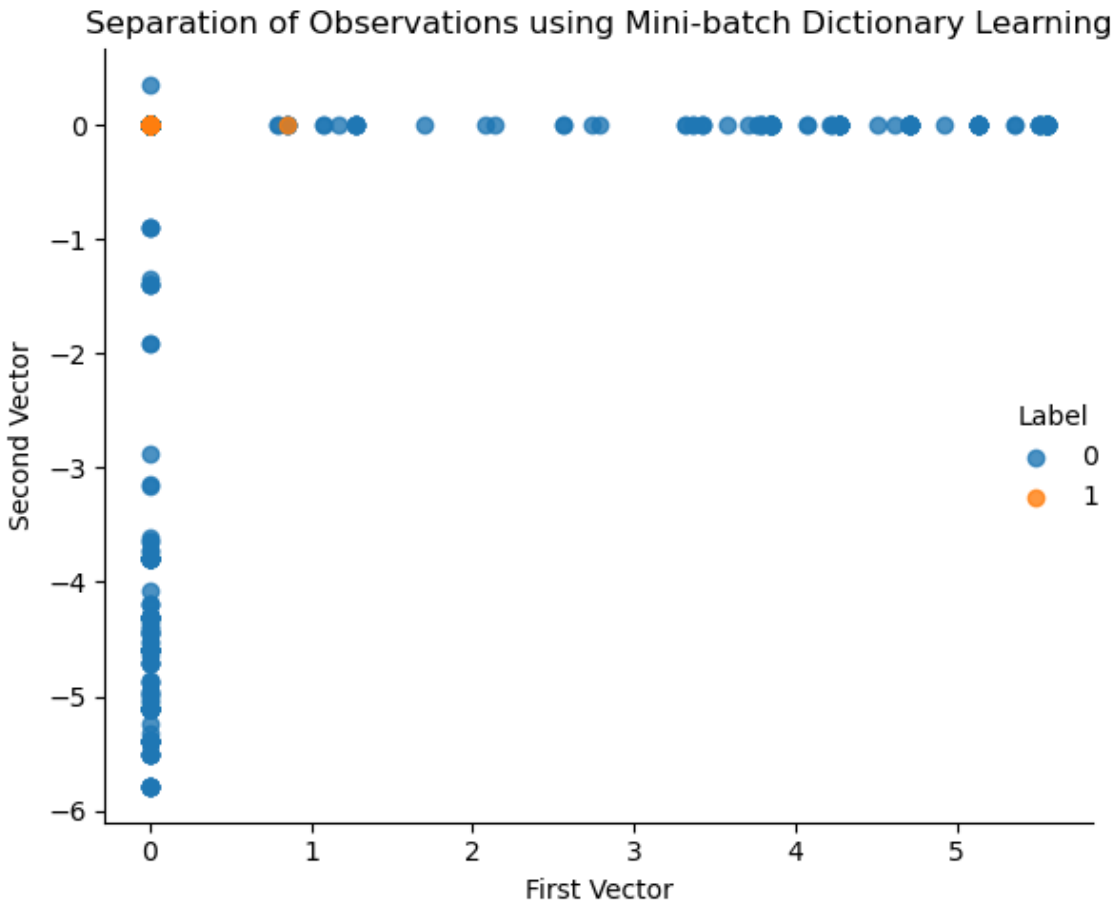
Cependant, en ajustant le nombre de composants à garder à 4, on obtient les mêmes performances que l'algorithme PCA :

```
Precision:  0.21
Recall:  0.3
Solar flares Caught out of 43 Cases: 9
```

Bien que l'utilisation de 4 composants principaux soit mieux, c'est loin d'être assez pour prédire correctement la plupart des instances d'éruptions solaires, nous devons donc continuer de chercher un algorithme de réduction de dimensionnalité qui nous donnera de bons résultats.

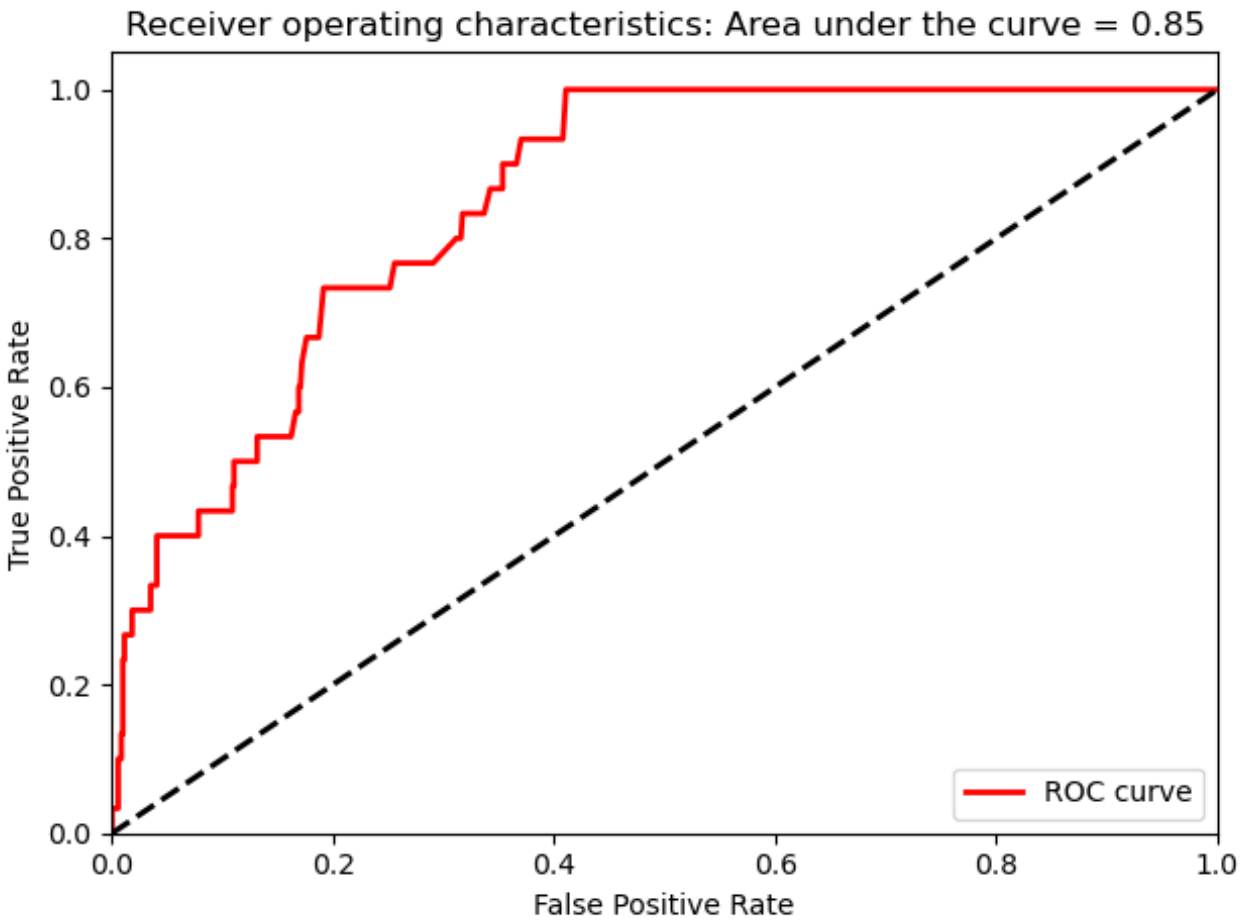
Apprentissage par dictionnaire

L'apprentissage par dictionnaire apprend la sparse représentation des données d'origine pour ensuite soit réduire le nombre de caractéristiques dans l'ensemble de données d'origine ou l'augmenter. Dans notre cas nous allons le réduire à 6 composants, comme dans les autres algorithmes. Ce type d'algorithme a deux paramètres que nous pouvons modifier afin d'essayer d'améliorer l'algorithme, ceux-ci sont le batch_size et le n_iter. Après plusieurs essais j'ai trouvé que les meilleurs paramètres étaient un batch_size de 15 et un n_iter de 1000. Voici les résultats :

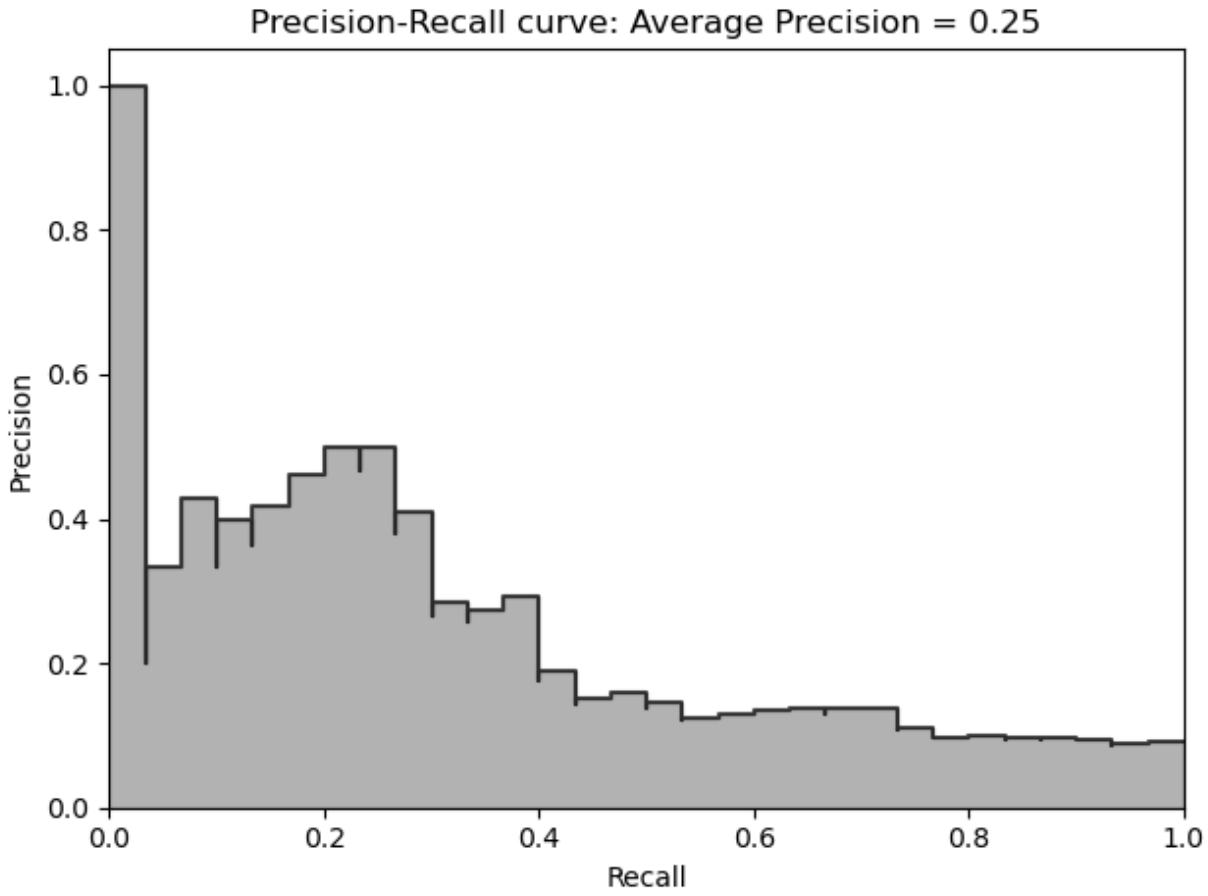


En premier lieu, on voit tout de suite que le graphique des instances avec les deux premiers composants est très différent des autres algorithmes, il suit deux axes et on peut aussi remarquer que les instances d'éruptions solaires sont tous rassemblées dans le haut à gauche. À première vue cet algorithme semble bien séparer les instances.

La courbe ROC aussi semble montrer que cet algorithme fonctionne bien avec l'ensemble de données, on y retrouve une aire sous la courbe de 0.85, ce qui est le plus élevé que nous avons jusqu'à présent.



On retrouve également la meilleur précision moyenne de tous les algorithmes que nous avons parcourus jusqu'à maintenant, elle est de 25%. On a détecté 40% des éruptions avec une précision de 40%.



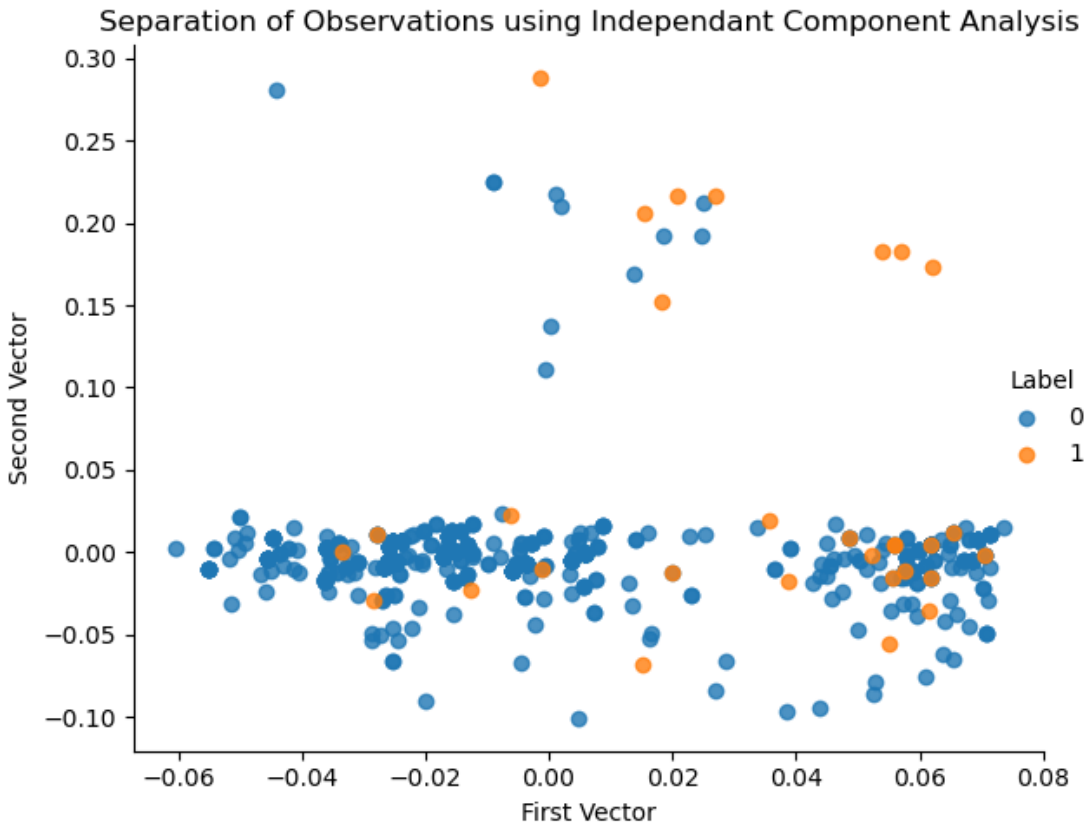

```
Precision:  0.28
Recall:  0.4
Solar flares Caught out of 43 Cases: 12
```

La modification du nombre de composants principaux ou du ratio de données de test diminue les performances du modèle, on les garde donc aux valeurs de 6 et 30%.

Cet algorithme de réduction de dimensionnalité est le meilleur jusqu'à présent, mais les résultats qu'il présente sont tout de même médiocre. Nous allons explorer un dernier algorithme avant de tester les meilleurs algorithmes avec les données de test.

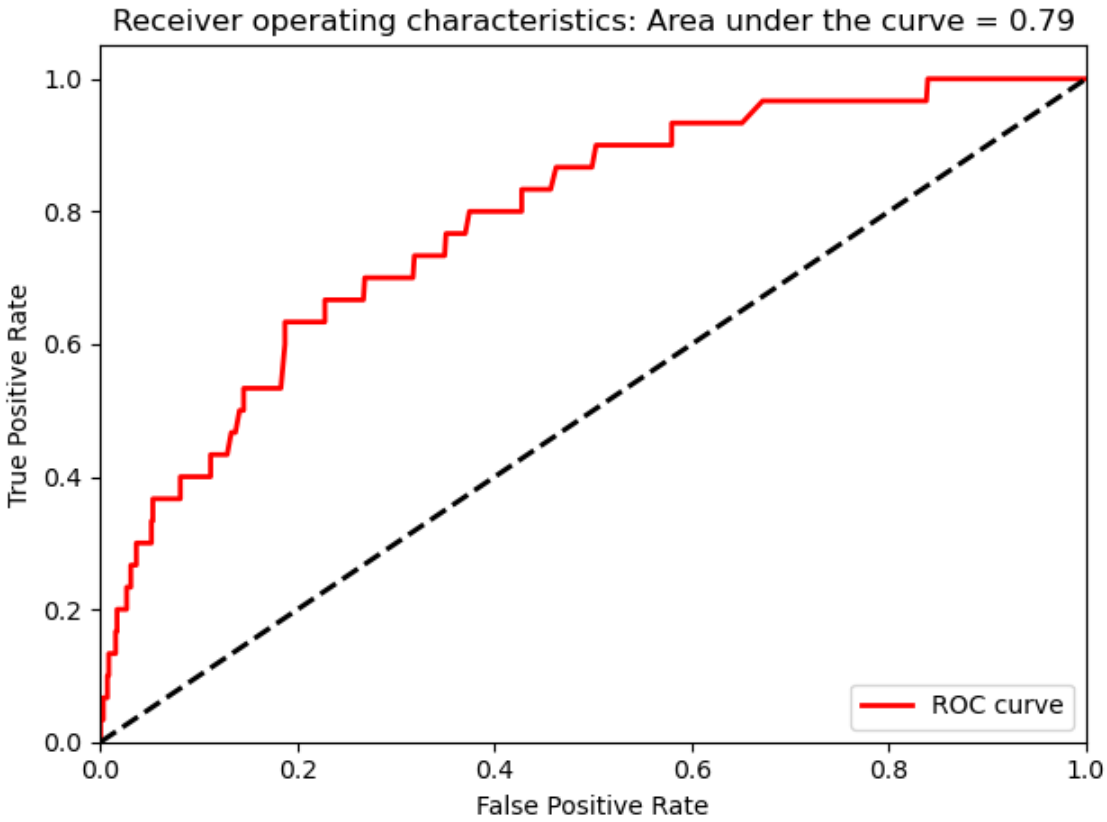
Analyse des composants indépendants

L'analyse des composants indépendants (ICA) sépare les signaux mélangés d'un ensemble de caractéristiques en signaux individuels. Nous allons l'utiliser pour détecter les éruptions solaires, comme d'habitude nous allons utiliser 6 composants principaux pour commencer. Voici les résultats :

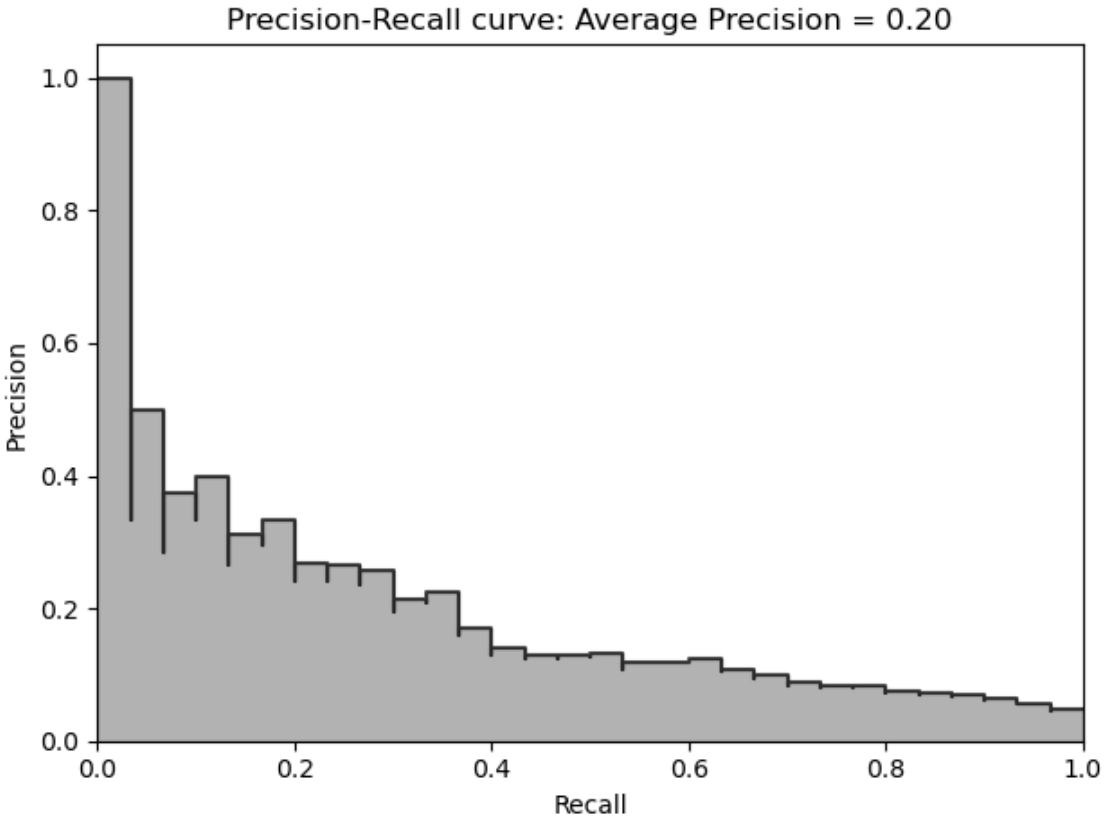


Dans le graphique des deux premiers composants principaux, on peut voir que deux clusters se forment dans le bas gauche et bas droit, mais ces clusters ne séparent pas les instances d'éruptions solaires des autres.

L’aire sous la courbe ROC est pareil à PCA, ce qui n’est pas trop mal.



Finalement, la précision moyenne de cet algorithme est de 20%, avec 30% des éruptions détectés et 21% de précision, ce qui est encore une fois pareil à PCA.



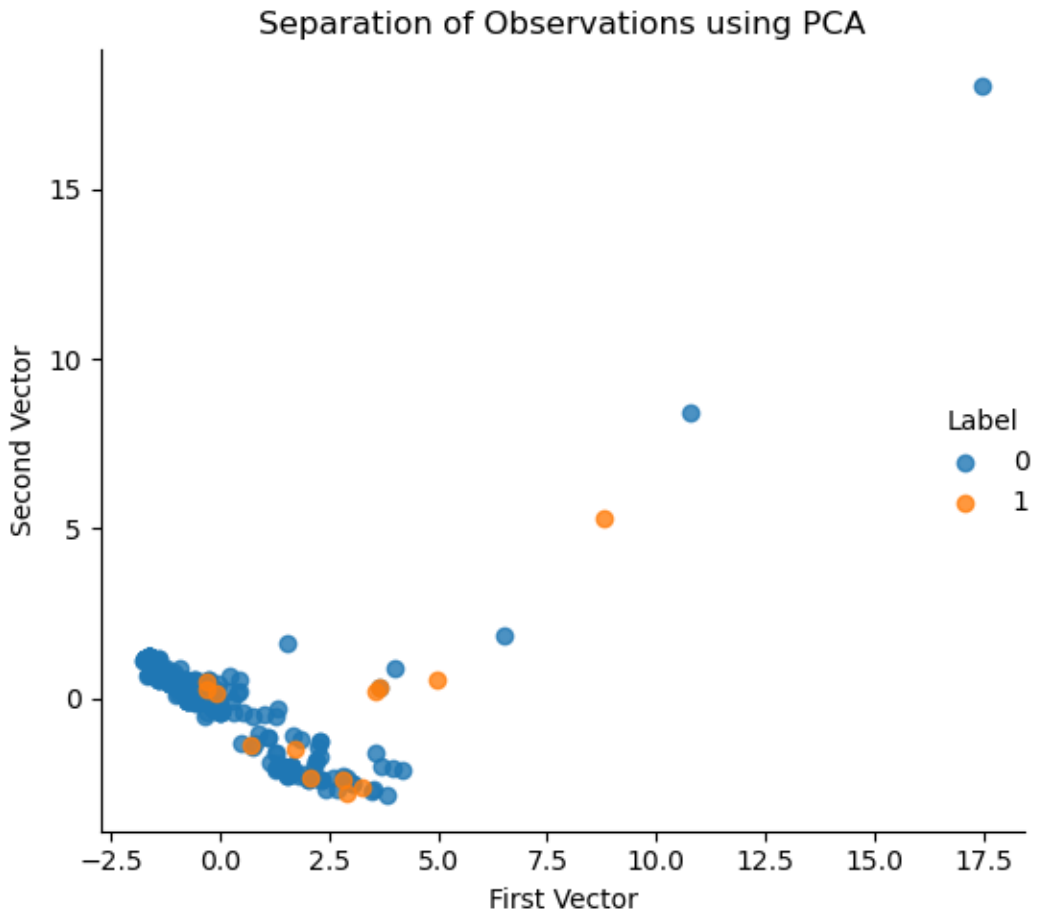

```
Precision:  0.21
Recall:  0.3
Solar flares Caught out of 43 Cases: 9
```

Les métriques nous permettent de voir que cet algorithme est très similaire à PCA en termes de performance, il est donc bon parmi nos algorithmes mais tout de même très médiocre. Maintenant que nous avons tester les algorithmes sur les données d'entrainement, faisons la même chose sur les meilleurs algorithmes sur les données de test.

Détection des éruptions sur l'ensemble de test

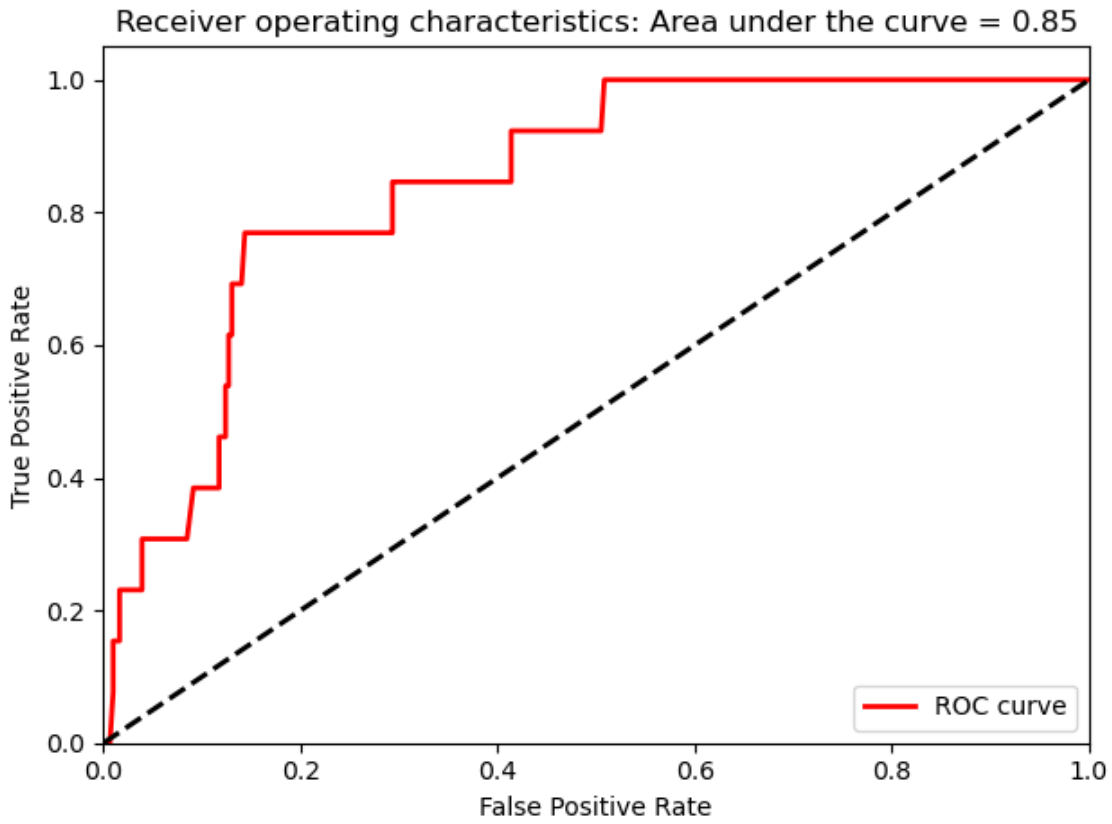
Nous allons tester les trois meilleurs algorithmes, soit PCA, Gaussian Random Projection et ICA.

Voici les résultats pour ce qui est de PCA :

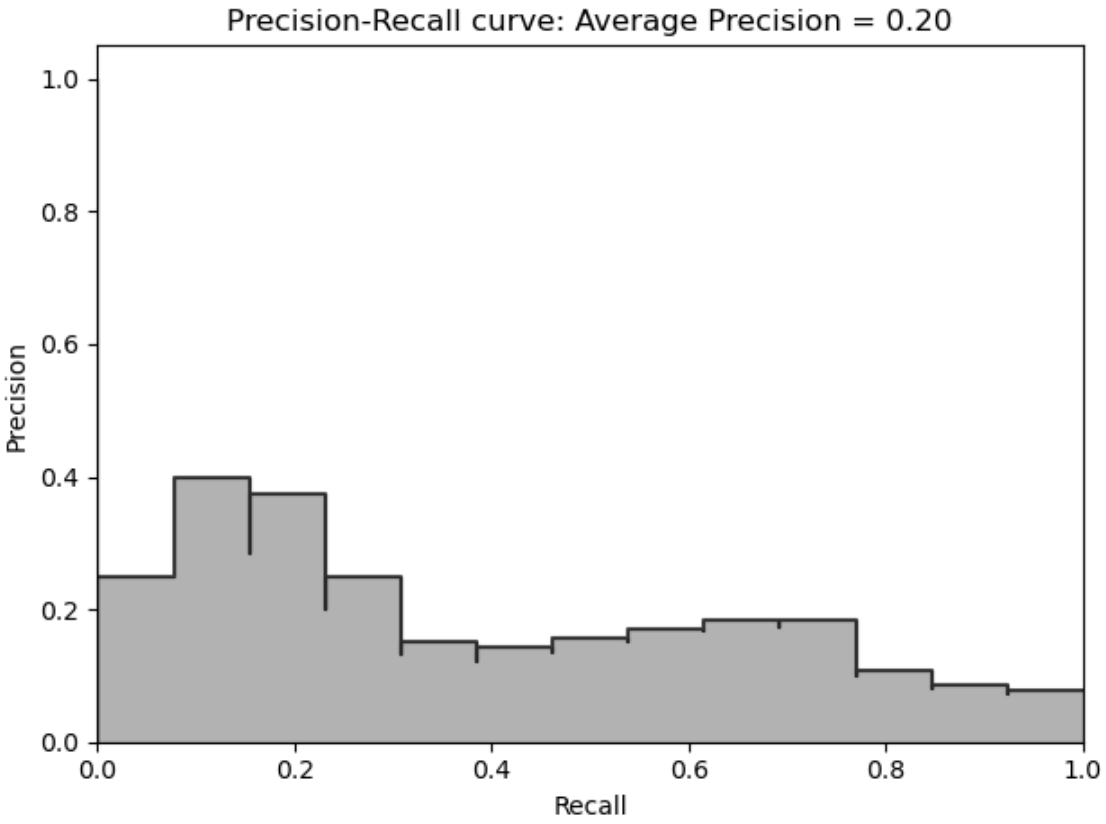


On peut voir qu'on forme un seul cluster moins éparpiller que celui de l'ensemble d'entrainement sur le graphique des instances avec les deux composants principaux.

L’aire sous la courbe ROC est meilleur de 0.05, cela égale notre meilleur résultat jusqu’à présent.



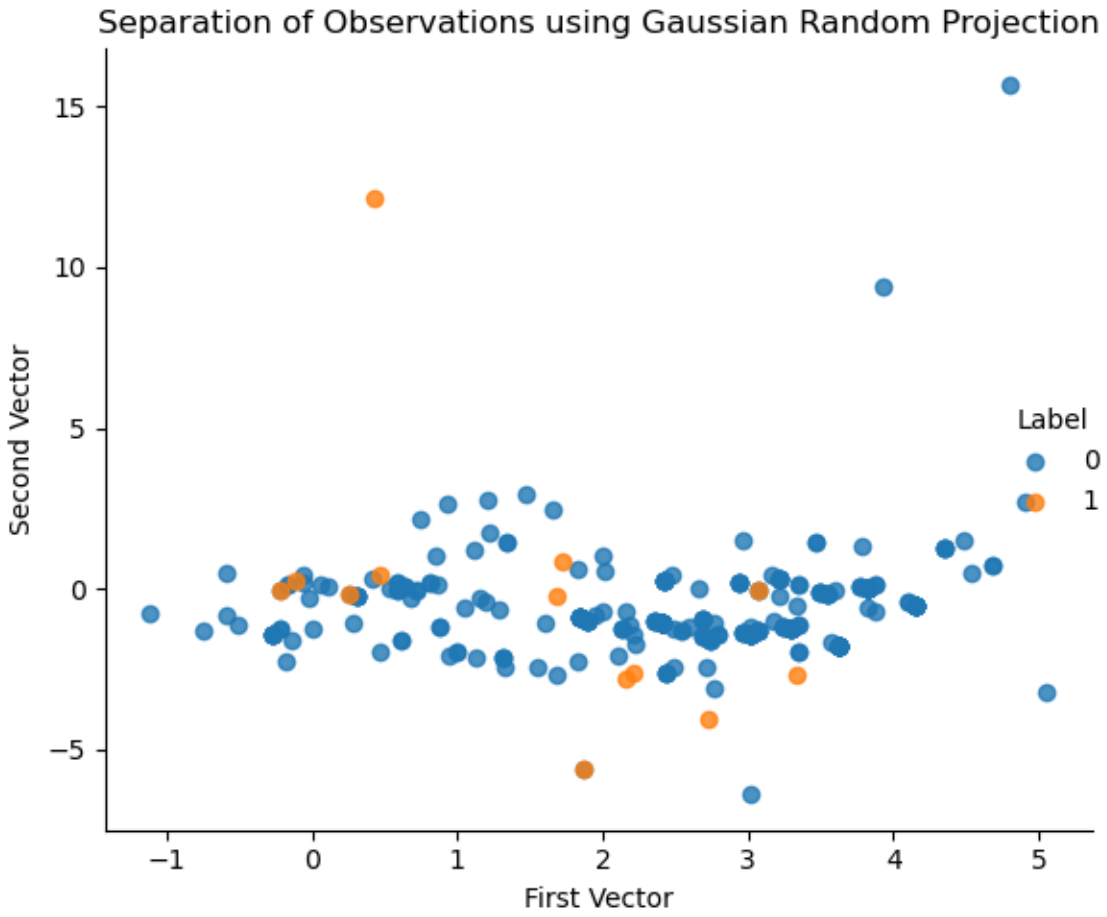
Le graphique précision-rappel nous montre une précision moyenne de 20%, 46% des instances d'éruptions solaires détectées avec 14% de précision. Cela est significativement moins bon qu'avec les données d'entrainement




```
Precision:  0.14
Recall:  0.46
Solar flares Caught out of 43 Cases: 6
```

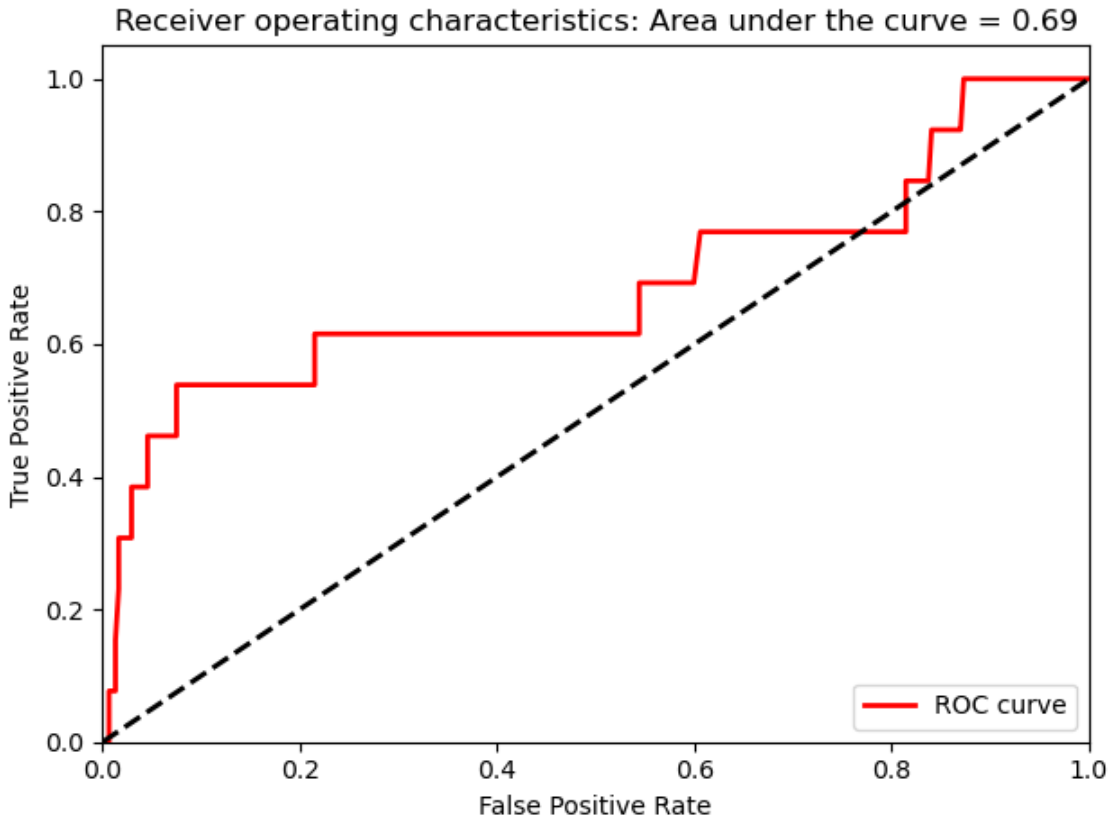
Selon ces métriques on pourrait dire que notre modèle n'est peut-être pas assez généralisé car il ne fonctionne pas aussi bien sur les données de test que sur les données d'entrainement.

Voici les résultats pour ce qui est de la Gaussian Random Projection :

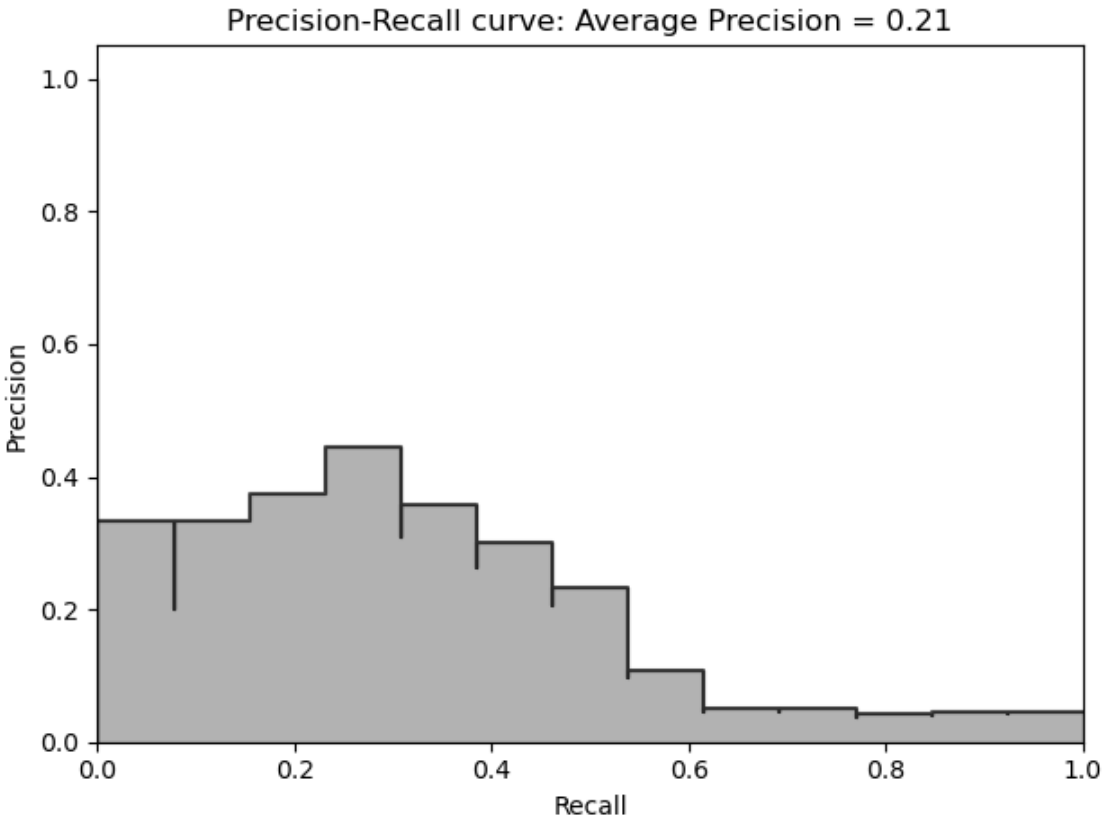


Le graphique des instances par rapport aux deux premiers composants est légèrement différent du premier mais on n'arrive pas non plus à distinguer les instances d'éruptions solaires de ceux qui ne le sont pas.

L’aire sous le courbe ROC est un peu mieux mais similaire à celle que l’on retrouve dans le modèle utilisant les instances d’entrainement.



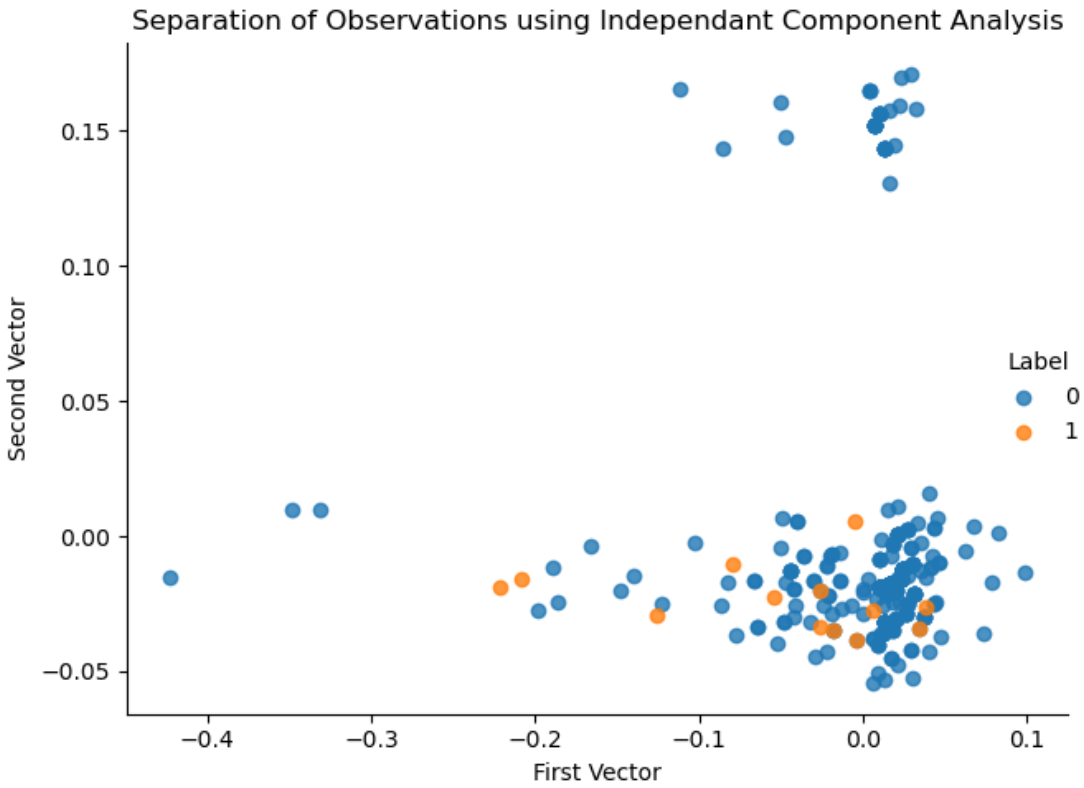
Finalement, la courbe précision-rappel et la précision est un peu mieux avec 54% des instances d'éruptions solaires détectées et 16% de précision mais tout de même mauvais.




```
Precision:  0.16
Recall:  0.54
Solar flares Caught out of 43 Cases: 7
```

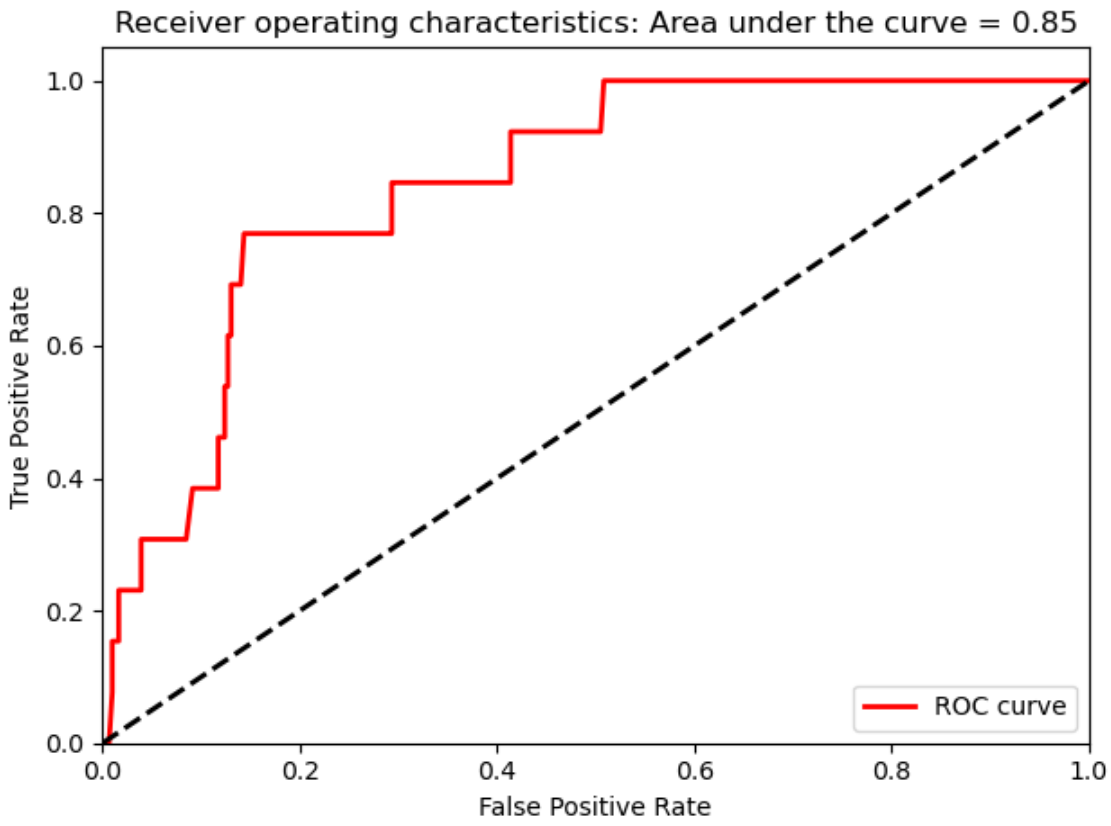
Les résultats avec les données de test sont un peu mieux mais loin d'être suffisant pour avoir un modèle qui peut réellement classer les instances d'éruptions solaires correctement.

Pour finir, voici les résultats de l'algorithme Fast ICA avec les données de test :

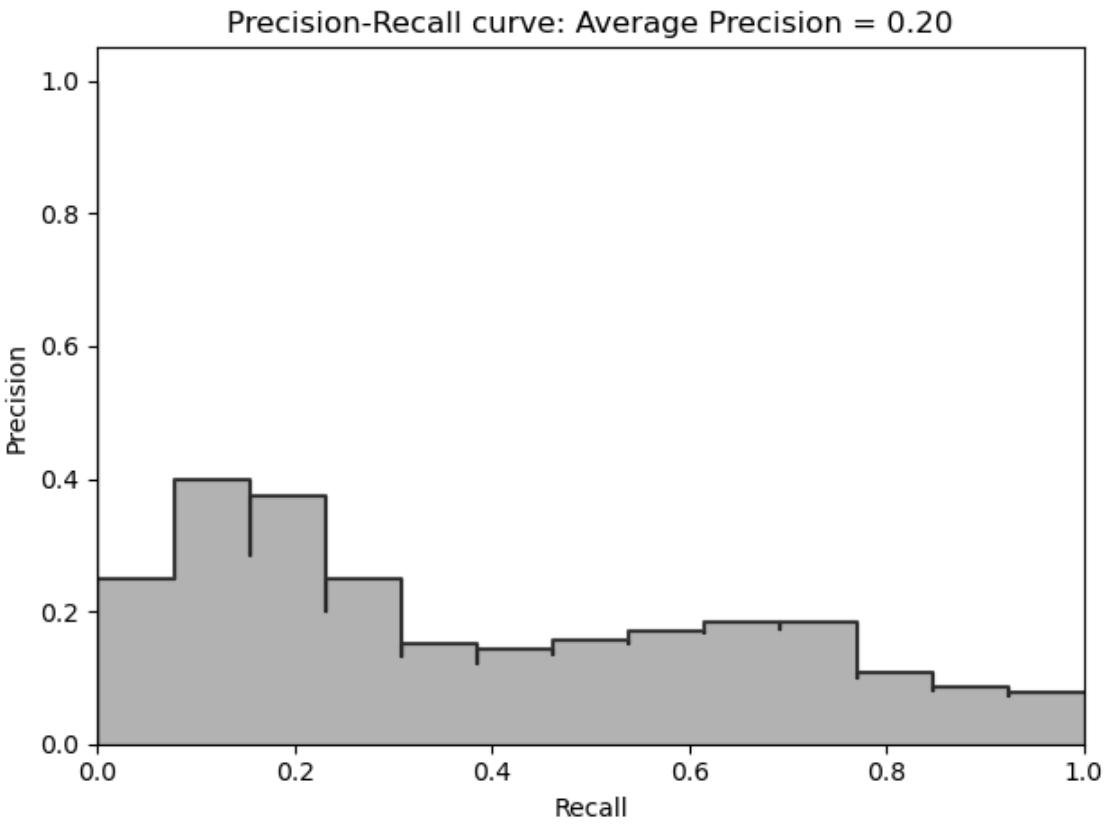


Contrairement aux résultats avec les données d'entrainement, on ne forme qu'un seul cluster, cependant les instances d'éruptions solaires sont toujours indistinguable des autres.

L’aire sous la courbe ROC est un peu mieux que celui que l’on retrouve sur le modèle avec les données d’entrainement.



La précision du modèle est meilleure avec les données de test, on a trouvé 46% des instances appartenant à la classe des éruptions solaires avec un précision de 14%.




```
Precision:  0.14
Recall:  0.46
Solar flares Caught out of 43 Cases: 6
```

Ces résultats sont très similaires que ceux que l'on a trouvés avec les données d'entrainement pour cet algorithme.

Conclusion de l'étude

Pour conclure cette étude, il est évident que les méthodes utilisées pour détecter les éruptions solaires dans cet ensemble de données ne sont pas adaptées.

Dans mon hypothèse de départ j'avais prédit que la précision moyenne des modèles ne dépasserait jamais 60% peu importe l'algorithme et j'avais également prédit qu'il serait impossible de distinguer les instances d'éruptions solaires des autres dans les graphiques dont les axes sont les deux premiers composants principaux. J'ai eu raison sur les deux points.

Je crois que certaines raisons peuvent expliquer la faible performance du modèle peu importe l'algorithme de réduction de dimensionnalité choisi. Le fait que l'ensemble de données ne possède que 11 attributs n'aident probablement pas à la réduction de dimensionnalité, je suis sûr que si l'on avait plus d'attributs on aurait pu avoir une meilleure vision de la différence entre les instances d'origine et celle reconstruites, ce qui nous aurait permis de mieux détecter les anomalies et du même coup, les éruptions solaires.

Un autre facteur est peut-être le fait que la plupart des attributs de l'ensemble de données sont catégoriels. Ces attributs sont des nombres entiers qui représente soit une catégorie ou un booléen. Je crois que les modèles que nous avons utilisés sont mieux adapté pour les ensembles de données comportant des nombres qui ne représentent pas des catégories, comme une vitesse ou une mesure.

On peut donc conclure de cette étude qu'aucun des algorithmes de réduction de dimensionnalité que nous avons essayés a été utiles et que nous devrions trouver d'autres méthodes pour détecter les éruptions solaires en utilisant cet ensemble de données.

Conclusion Générale

Pour finir, ce travail m'as permis de vraiment mieux comprendre comment la détection d'anomalie fonctionnait et comment on peut s'en servir dans des applications réelles malgré l'échecs de cette méthode sur l'ensemble de données choisi.

Les visualisations suggérées dans le travail sont très utiles et le fait que j'ai dû faire quelques recherches sur Internet m'as permis de consolider ma compréhension de ces visualisations, surtout la courbe ROC et l'aire sous cette courbe.

Je crois que j'aurais dû choisir un ensemble de données qui ne comportait aucun attribut catégoriel car cela semble donner de mauvais résultats.