Randrianarisona Aaron

Bellion Antoine

Rapport de projet : estimation de la satisfaction de patients vis-à-vis des médicaments

I.Description du jeux de données

1.Contexte / problématique

Dans le cadre du suivi de la qualité des produits pharmaceutiques, les professionnels peuvent soumettre les produits à une évaluation auprès des patients. Ceux-ci sont encouragés à émettre un avis comprenant un descriptif textuel de leur expérience avec le traitement, ainsi qu'un score de satisfaction allant de 1 à 10.

Ainsi, on obtient un jeu de données répertoriant les avis des patients, incluant le nom du médicament, la condition médicale ayant conduit à la prise du traitement, ainsi que le contenu de l'avis accompagné de sa date d'émission.

Étant donné la taille importante de ce jeu de données, on souhaite utiliser des techniques de machine learning pour :

- Développer un modèle capable de prédire un score de satisfaction entre 1 et 10 à partir du commentaire d'un patient.
- Concevoir un modèle qui attribue une classe (négatif, neutre, positif) à un médicament.

2. Statistiques

Quelques statistiques intéressantes concernant le jeu de données :

- Les intitulés du dataset : ['Unnamed: 0', 'patient_id', 'drugName', 'condition', 'review', 'rating', 'date', 'usefulCount', 'review_length']
- Nombre total d'avis : 46 108
- Nombre total de médicaments : 2 283
- Moyenne d'avis par médicament : 20,20

Nombre d'avis par rating et aperçu du dataset :

rating	count
10.0	14276
9.0	7972
1.0	6034
8.0	5423
7.0	2738
5.0	2394
2.0	2028
3.0	1931
6.0	1860
4.0	1452

	Unnamed: 0	patient_id	drugName	condition	review	rating	date	usefulCount	review_length
0	0	163740	Mirtazapine	depression	"i've tried a few antidepressants over the yea...	10.0	February 28, 2012	22	68
1	1	206473	Mesalamine	crohn's disease, maintenance	"my son has crohn's disease and has done very ...	8.0	May 17, 2009	17	48
2	2	39293	Contrave	weight loss	"contrave combines drugs that were used for al...	9.0	March 5, 2017	35	143

II. Algorithmes et Technologies (Etat de l'Art)

Dans cette section, nous allons explorer différentes techniques et algorithmes utilisés pour le traitement du langage naturel (TNL) et l'analyse prédictive dans le domaine des avis clients et des opinions des patients. Les principales méthodes incluent la vectorisation du texte, l'analyse des sentiments, ainsi que les techniques de classification et de régression en machine learning, avec un focus particulier sur les approches classiques et de pointe. Enfin, nous aborderons l'utilisation des arbres de décision pour des tâches de prédiction et de classification.

1.Vectorisation du texte:

La vectorisation du texte est une étape cruciale dans le traitement du langage naturel (TNL), car elle permet de convertir des données textuelles en un format numérique compréhensible par les modèles d'apprentissage automatique. Plusieurs techniques de vectorisation sont utilisées dans le domaine du traitement automatique des langues :

- **Bag-of-Words (BoW)** : Cette méthode consiste à représenter chaque document par un vecteur de fréquence d'apparition des mots dans le texte. Bien qu'efficace, elle ne capture pas les relations entre les mots.
- **TF-IDF (Term Frequency - Inverse Document Frequency)** : Une amélioration du modèle BoW qui prend en compte l'importance des termes dans un document par rapport à l'ensemble du corpus. Cela permet de donner plus de poids aux mots rares et significatifs.
- **Word Embeddings (Word2Vec, GloVe)** : Ces modèles de vectorisation représentent les mots par des vecteurs denses de faible dimension. Ils capturent les relations sémantiques entre les mots, ce qui permet de mieux modéliser les synonymes ou les mots ayant un sens similaire dans différents contextes.
- **Transformers et BERT** : Les modèles de type transformer, comme BERT (Bidirectional Encoder Representations from Transformers), permettent une représentation contextuelle des mots, où chaque mot est compris en fonction de son environnement dans la phrase. Cette technique est particulièrement performante pour des tâches complexes comme la traduction, la question-réponse, et l'analyse de sentiment.

Applications :

- Préparation des données textuelles pour les étapes d'analyse de sentiments, de classification ou de clustering.
- Utilisation de Word2Vec ou GloVe pour améliorer les résultats des modèles en capturant mieux les relations sémantiques dans des avis ou des commentaires.

2. Analyse et prédiction de sentiment des avis utilisateur

L'analyse de sentiment vise à déterminer l'attitude d'un auteur à l'égard d'un sujet donné, souvent en classifiant les opinions en catégories comme "positif", "négatif" ou "neutre". Cette analyse est particulièrement utilisée pour examiner les avis des clients ou des patients dans

des contextes tels que la gestion de la réputation, la santé publique ou l'expérience utilisateur.

- **Méthodes basées sur des lexiques** : Ces approches utilisent des dictionnaires de mots avec des scores associés pour déterminer le sentiment général d'un texte. Exemple : le lexique SentiWordNet.
- **Modèles supervisés (SVM, Naive Bayes, etc.)** : Après la vectorisation du texte, des modèles classiques tels que les SVM (Support Vector Machines) ou Naive Bayes peuvent être utilisés pour classer les avis selon leur polarité (positive, négative, neutre).
- **Approches profondes avec RNN et LSTM** : Les réseaux de neurones récurrents (RNN), et en particulier les Long Short-Term Memory (LSTM), sont souvent utilisés pour les tâches de prédiction de sentiment sur des données textuelles. Ces modèles sont capables de capturer les dépendances temporelles dans le texte, ce qui est utile pour analyser des phrases longues.
- **Modèles Transformer (BERT, RoBERTa, etc.)** : L'utilisation de modèles pré-entraînés comme BERT a révolutionné l'analyse de sentiment. Ces modèles permettent de traiter des données textuelles de manière contextuelle, en capturant plus précisément les nuances du langage humain.

Applications :

- Analyse de la satisfaction des clients à partir des avis en ligne.
- Évaluation du sentiment des patients concernant des traitements médicaux ou des consultations via des enquêtes ou des plateformes de feedback.

3. Méthodes de classification et de régression en machine learning : Revue des approches classiques et des techniques de pointe

Les modèles de classification et de régression sont des outils essentiels pour prédire des résultats à partir de données textuelles ou numériques. Ces techniques sont largement utilisées dans l'analyse des avis clients, l'identification des tendances et la prédiction d'événements futurs.

3.1 Méthodes classiques

- **SVM (Support Vector Machines)** : Très populaire dans des tâches de classification binaire ou multi-classe, les SVM sont utilisés pour séparer des classes distinctes à l'aide d'un hyperplan. Cette approche est efficace, notamment dans les cas où les données sont non linéaires, en utilisant des noyaux adaptés.
- **K-NN (K-Nearest Neighbors)** : Cette méthode consiste à classer un point de données en fonction des labels de ses voisins les plus proches dans l'espace des caractéristiques. Bien que simple, elle peut être gourmande en ressources et inefficace sur des ensembles de données volumineux.

- **Arbres de décision et forêts aléatoires** : Les arbres de décision permettent de diviser un espace de décision en fonction de certaines caractéristiques, tandis que les forêts aléatoires combinent plusieurs arbres pour améliorer la robustesse et éviter le surapprentissage.

3.2 Techniques de pointe

- **Réseaux de neurones (Deep Learning)** : L'usage des réseaux de neurones profonds (DNN) permet de capturer des relations complexes dans les données. Ces modèles sont particulièrement performants pour des tâches non linéaires et à grande échelle.
- **Transformers** : Ces modèles, en particulier BERT, GPT, et leurs variantes, ont marqué un tournant dans l'analyse de texte en raison de leur capacité à traiter de grandes quantités de données avec une grande précision. Ils permettent de traiter des tâches complexes telles que la traduction automatique, la réponse aux questions, et l'analyse de sentiment.
- **Apprentissage en renforcement** : Bien que moins couramment utilisé dans le traitement du texte, l'apprentissage en renforcement est de plus en plus exploré pour des tâches comme la personnalisation des avis clients et la recommandation automatique.

Applications :

- Prédiction de la satisfaction des clients ou de l'engagement des patients en fonction de leurs réponses dans les enquêtes.
- Classification automatique des textes dans des catégories (avis positifs, négatifs, neutres).

III. Réalisations

1. Modèle de prédiction de score

a) Pré-traitement

Dans cette partie, nous souhaitons nous concentrer uniquement sur deux caractéristiques : la colonne 'review' et la colonne 'rating'.

- **Vectorisation**
  Afin de préparer les jeux de données pour l'apprentissage du modèle de prédiction de score basé sur les commentaires des patients, il est nécessaire de transformer ces commentaires textuels en une représentation vectorielle, un procédé appelé *word embedding*. Ici, chaque commentaire est un paragraphe, constitué de plusieurs phrases qui apportent progressivement un contexte et construisent un sentiment. Dans ce contexte, l'utilisation du modèle Sentence-BERT est particulièrement pertinente, car lors de la vectorisation, il capture le contexte de chaque commentaire, ce qui le rend bien adapté pour l'analyse de sentiments dans les avis patients.
- **Séparation des ensembles**
  Nous préparons ensuite les ensembles d'entraînement et de test à partir des

commentaires des patients vectorisés, avec leurs scores réels associés. En respectant un ratio de 80 % des données pour l'apprentissage et 20 % pour l'évaluation, les données sont sélectionnées aléatoirement pour maintenir la proportion. La variable `RANDOM_SEED` est utilisée pour conserver la répartition des données.

- **Standardisation**
  Nous appliquons une standardisation des variables d'entrée en utilisant `StandardScaler` pour centrer les données sur une moyenne de 0 avec un écart-type de 1. Cela permet d'éviter que des valeurs élevées de certaines caractéristiques dominent le modèle, améliorant ainsi les performances.

b) Apprentissage des modèles:

Pour chaque modèle, nous avons suivi la même démarche : apprentissage sur l'ensemble d'entraînement (ici, les vecteurs des commentaires textuels des patients), exploration rapide des hyperparamètres, et ajustement de certaines valeurs pour observer leur impact sur les performances, en particulier lorsque le modèle s'entraîne relativement rapidement.

Les modèles suivants ont été entraînés :

- Random Forest Regressor
- Régression linéaire
- Support Vector Regression
- Decision Tree Regression
- Un réseau de neurones avec une couche de régression

Note : Nous avons tenté d'optimiser les hyperparamètres pour chaque modèle afin d'obtenir les meilleures performances possibles, en utilisant l'API `GridSearchCV`. Cependant, nous avons rencontré un problème de chargement infini lors de l'entraînement, sans retour d'avancement. Néanmoins, `GridSearchCV` a pu fournir des résultats pour le modèle Decision Tree.

- Decision Tree Regression

```python
from sklearn.tree import DecisionTreeRegressor

param_grid = {
    'max_depth': [None, 5, 10, 20],
    'min_samples_split': [2, 5, 10],
    'min_samples_leaf': [1, 2, 5],
    'max_features': ['sqrt', 'log2']
}

grid_search = GridSearchCV(estimator=DecisionTreeRegressor(random_state=RANDOM_SEED),
                           param_grid=param_grid, cv=5,
                           scoring='neg_mean_squared_error', n_jobs=-1, verbose=3)
grid_search.fit(X_train_review, y_train_rating)

dtr_best_params = grid_search.best_params_
dtr_model = grid_search.best_estimator_
print("Best Hyperparameters:", dtr_best_params)
```

```
Fitting 5 folds for each of 72 candidates, totalling 360 fits
Best Hyperparameters: {'max_depth': 5, 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_samples_split': 2}
```

c) Evaluations et comparaison des modèles

Nous utilisons les modèles entraînés précédemment pour prédire les scores de satisfaction des patients à partir de l'ensemble de données réservé pour le test et l'évaluation.

- Voici un aperçu des valeurs prédites par chaque modèle pour des avis sélectionnés aléatoirement:

	avis client	Score Réel	Random Forest	Régression Linéaire	SVR	Decision Tree	Neural Network
32965	"i have been on nuvaring for 5 months now. i h...	5.0	6.486667	6.488390	6.287305	6.464605	6.649881
37593	"bonine is a miracle if you are bothered by mo...	10.0	7.500000	11.550209	9.707582	8.047011	10.751434
41987	"i started taking abilify about 2 weeks ago fo...	8.0	7.050000	8.646396	8.065429	8.356050	6.848653
22215	"citalopram changed my life. i started taking ...	10.0	8.190000	8.357454	9.061120	6.710648	9.639012
26778	"i have just started taking contrave and am on...	4.0	7.170000	6.530591	6.627708	7.830000	6.724073
648	"it works! i took tindamax 2 day dosing, 4 pil...	10.0	4.950000	6.025125	6.840080	5.733058	5.041559
13632	"i got up sunday morning with a cold sore. did...	10.0	6.750000	7.225096	8.318875	6.844199	9.159936
2740	"i have been on kariva for about five months. ...	9.0	7.890000	7.592611	8.699768	6.710648	8.403659
41835	"i have tried everything the only one that rea...	10.0	8.600000	9.081514	9.672858	6.710648	8.942764
13128	"i woke up this past thursday with food poison...	10.0	5.470000	5.968337	6.708787	5.733058	7.569873

- Les prédictions pour des score de satisfaction initiales élevées:

	avis client	Score Réel	Random Forest	Régression Linéaire	SVR	Decision Tree	Neural Network
20222	"i've been on microgestin for about 6 months a...	10.0	7.60	9.584156	9.140924	8.047011	10.564593
23503	"hello, i started taking contrave april 18,201...	10.0	6.48	7.359789	8.148883	7.741689	9.632801
24639	"i started topamax in 2005. doctor said he was...	10.0	6.57	6.131080	5.661510	5.733058	4.489933
43032	"hi i've been on orlistat now for 4 weeks had ...	10.0	6.77	7.764977	9.353950	7.481047	9.092247
21444	"i had psoriasis come up suddenly all over my ...	10.0	6.39	7.362596	8.272439	5.864675	7.371913
5510	"this is my third type of birth control, and i...	10.0	7.30	6.685172	6.509958	7.741689	5.699075
3007	"i started safyral in april for irregular mens...	10.0	7.52	6.767808	7.107434	6.292443	7.425599
17639	"i without a doubt believe this to be the best...	10.0	7.99	8.551741	9.567852	5.094138	12.022900
8934	"i started this about a week and half ago and ...	10.0	8.98	8.173923	8.933316	7.785536	8.074881
7778	"i had four polyps removed two years ago. my p...	10.0	7.27	7.766353	9.440322	6.595070	9.784141

- Les prédictions pour des scores de satisfaction initiales faibles:

	avis client	Score Réel	Random Forest	Régression Linéaire	SVR	Decision Tree	Neural Network
39246	"i had the paraguard put in by my midwife, who...	1.0	3.96	3.885498	3.873793	5.733058	0.383991
6029	"had 1st shot of simponi and didn't notice any...	1.0	6.85	4.670197	5.579917	5.094138	2.893344
14229	"i would never recommend implanon (the rod) as...	1.0	3.25	7.875805	8.325815	6.710648	5.042061
11711	"i was on depot for 4 months after giving birt...	1.0	4.42	3.753028	4.735840	6.292443	4.447931
39829	"in 3 days of use i had itchy scalp within 2 w...	1.0	6.43	4.053479	5.451452	5.733058	2.881758
45847	"well, it seemed like a good decision at the t...	1.0	6.55	8.659279	9.955096	7.481047	7.680606
37535	"i was put on this bc to prevent pregnancy. i ...	1.0	4.87	5.447325	3.866356	6.524522	4.082644
44356	"worst pill ever, i had evasive thoughts. i wo...	1.0	3.13	3.338263	2.386066	5.094138	2.462887
36687	"i took 1 30mg capsule of cymbalta yesterday. ...	1.0	4.42	5.606674	4.506520	6.204114	4.197266
5775	"i'm a very active 63 yr old with a 4x bypass....	1.0	7.44	6.488978	7.188781	6.464605	6.098766
17102	"i am usually not the type to comment or write...	1.0	5.36	3.667820	3.523201	5.094138	-0.920093
16274	"my doctor switched me from ms cotin 10 mg 3 t...	1.0	5.95	4.140683	2.745262	6.770979	4.280954
44115	"worst bc i've ever taken. i thought it be nic...	1.0	5.87	4.989134	3.915367	4.784387	6.218041
7402	"i went to my pain management doctor today & h...	1.0	6.83	5.646525	7.498380	6.770979	7.512871
25975	"dosage was 500 mg for 10 days. stopped after ...	1.0	5.90	6.599148	6.643404	7.562044	5.013699
5977	"my experience with nucynta was very dis-appoi...	1.0	7.62	5.778006	7.125065	6.595070	7.022406
18106	"i had a bad experience with contrave. first d...	1.0	1.69	1.473206	0.983138	6.362631	0.951592

c.a) Comparaison quantitative

Il existe plusieurs métriques pour évaluer les performances d'un modèle. Pour la prédiction de scores, donc dans un contexte de régression, nous utiliserons les métriques les plus courantes suivantes :

- **Mean Absolute Error (MAE)**
  La moyenne des erreurs absolues, qui mesure l'écart moyen entre les prédictions et les valeurs réelles. Plus la MAE est faible, meilleure est la précision du modèle.
- **Root Mean Squared Error (RMSE)**
  La racine carrée de la moyenne des erreurs quadratiques. Cette métrique accentue les grandes erreurs (dues au carré des différences) et pénalise davantage les prédictions très éloignées des valeurs réelles.
- **R^2 (coefficient de détermination)**
  Il indique la proportion de la variance des scores réels expliquée par les prédictions du modèle. Un R^2 proche de 1 signifie un bon ajustement, tandis qu'une valeur proche de 0 indique un faible ajustement.

Nous obtenons le tableau comparatif suivant :

	Metric	Random Forest	Régression Linéaire	SVR	NNR
0	MAE	2.183836	2.180532	1.913344	2.099844
1	RMSE	2.690243	2.692240	2.560549	2.676334
2	R2	0.329257	0.328261	0.392370	0.336175

Sur la base de ces résultats, nous pouvons en déduire que, dans notre cas, le modèle le plus performant est le modèle SVR, tandis que le moins performant est le modèle de régression linéaire.

c.b) Analyse qualitative

Une analyse possible consiste à déterminer, pour chaque modèle, s'il a tendance à surestimer ou sous-estimer la valeur initiale. Pour cela, nous calculerons la moyenne des différences entre le score prédit et le score réel.

Voici les résultats obtenus :

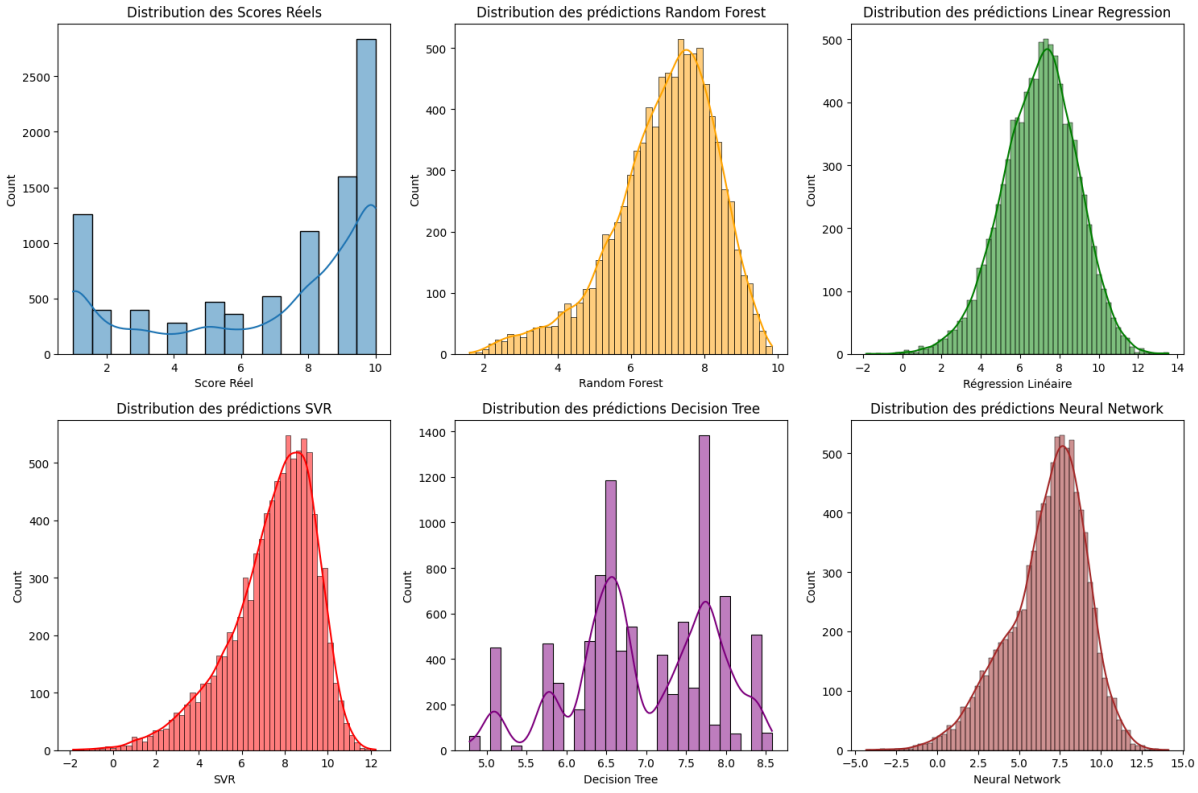
```
Statistiques sur les erreurs de predictions:
          Diff_RFR  Diff_LinReg     Diff_SVR     Diff_DTR     Diff_NNR
count  9222.000000  9222.000000  9222.000000  9222.000000  9222.000000
mean     -0.028298     0.040001     0.466044     0.022182     0.009129
std       2.690240     2.692088     2.517916     3.185025     2.663485
min      -6.010000   -10.327541   -10.441779    -5.215613   -12.623962
25%      -1.987500    -1.950357    -1.186644    -2.404930    -1.709138
50%      -0.670000    -0.370220     0.011927    -1.214464    -0.266538
75%       1.640000     1.818336     1.861669     2.595070     1.540609
max       8.210000    12.551895     9.565078     7.575540    10.247353

Random Forest mean difference: -0.0282980915202776
Random Forest a tendance à sous-estimer

Linear Regression mean difference: 0.040000666146990896
LinReg a tendance à surestimer

SVR mean difference: 0.4660440164531707
SVR a tendance à surestimer

Decision Tree mean difference: 0.022182256565835484
DTR a tendance à surestimer

Neural Network mean difference: 0.009129027231503299
Neural Network a tendance à surestimer
```

Il est également intéressant d'examiner le biais des valeurs extrêmes, en observant s'il y a une surestimation ou une sous-estimation plus fréquente pour les valeurs fortes ou faibles. Nous pouvons utiliser un masque basé sur les seuils négatif, neutre et positif sur l'ensemble

de test pour diviser celui-ci en trois groupes correspondant à ces catégories.

```
RFR:
  Moyenne des differences pour les scores élevés (>= 7): -1.6327265818602346
  RFR a tendance à sous-estimer les scores élevés.
  Moyenne des differences pour les scores faibles(<= 4): 3.711787574722459
  RFR a tendance à surestimer scores faibles

LinReg:
  Moyenne des differences pour les scores élevés (>= 7): -1.3876927879336016
  LinReg a tendance à sous-estimer les scores élevés.
  Moyenne des differences pour les scores faibles(<= 4): 3.4035579827379507
  LinReg a tendance à surestimer scores faibles

SVR:
  Moyenne des differences pour les scores élevés (>= 7): -0.8265014025927029
  SVR a tendance à sous-estimer les scores élevés.
  Moyenne des differences pour les scores faibles(<= 4): 3.520288852702145
  SVR a tendance à surestimer scores faibles

DTR:
  Moyenne des differences pour les scores élevés (>= 7): -2.0030279374613897
  DTR a tendance à sous-estimer les scores élevés.
  Moyenne des differences pour les scores faibles(<= 4): 4.7393016518748645
  DTR a tendance à surestimer scores faibles

NNR:
  Moyenne des differences pour les scores élevés (>= 7): -1.4464430954807674
  NNR a tendance à sous-estimer les scores élevés.
  Moyenne des differences pour les scores faibles(<= 4): 2.5959336389775567
  NNR a tendance à surestimer scores faibles
```

Nous pouvons aussi visualiser la distribution des prédictions par rapport aux valeurs réelles sous forme d'histogrammes.



Interprétation: La plupart des modèles se concentrent autour des valeurs moyennes, montrant une difficulté à prédire correctement les valeurs extrêmes, en particulier les faibles valeurs. Cela pourrait s'expliquer en partie par la domination des valeurs supérieures à 8 dans le jeu de données, ce qui influence l'apprentissage.

2. Modèle de classification

a) Pré-traitement

Dans cette partie, les caractéristiques (features) qui vont nous intéresser seront le nom du médicament ("drugName") et le score de l'avis de satisfaction du patient ("rating").

Contrairement à l'étape précédente, où nous avons utilisé des textes longs à encoder, ici nous n'avons que le nom du médicament. Cependant, comme les noms de médicaments peuvent être composés de plusieurs mots et inclure des traitements conjoints, l'encodage basé sur l'embedding semble plus adapté. Nous avons choisi d'utiliser `CountVectorizer` pour transformer les noms des médicaments en vecteurs.

Il faut ensuite définir les seuils qui permettront de décider si un médicament doit être classé comme Négatif, Neutre ou Positif. Nous avons choisi les seuils suivants :

- Négatif : 1 -> 4
- Neutre : 5 -> 6
- Positif : 7 -> 10

b) Apprentissage des modèles

Pour chaque modèle, nous avons adopté la même démarche : apprentissage sur l'ensemble d'entraînement (ici, les vecteurs des noms des médicaments), exploration rapide des hyperparamètres du modèle, en ajustant les valeurs de certains paramètres pour observer l'impact sur la performance, lorsque l'entraînement du modèle est relativement rapide.

Les modèles suivants ont été entraînés :

- SVM
- Régression logistique
- Forêt aléatoire
- Algorithme basé sur la moyenne des scores de satisfaction
- Algorithme basé sur la classe majoritaire

NB : Ici, le paramètre `class_weight` est recommandé en raison du déséquilibre dans la répartition des scores dans le jeu de données, majoritairement localisé dans le seuil positif.

c) Evaluations et comparaison des modèles

Nous utilisons les modèles entraînés précédemment pour prédire la classe du médicament à partir de l'ensemble de test et évaluer les performances des modèles.

- Voici un aperçu des valeurs prédites pour un échantillon aléatoire :

	Nom du médicament	Classe réelle	Random Forest	Régression Logistique	SVM	Classe Prédite (Moyenne)	Classe Prédite (Majorité)
8861	Mononessa	positive	neutral	neutral	neutral	neutral	positive
2906	Methadone	negative	positive	positive	positive	positive	positive
4103	Hyoscyamine / methenamine / methylene blue / p...	positive	neutral	neutral	neutral	positive	positive
8637	Lo Loestrin Fe	positive	negative	negative	negative	neutral	positive
1448	Lo Loestrin Fe	negative	negative	negative	negative	neutral	positive
6268	Avapro	positive	neutral	neutral	neutral	positive	positive
3252	Yaz	negative	negative	negative	negative	neutral	positive
4783	Amphetamine / dextroamphetamine	positive	positive	positive	positive	positive	positive
5334	Drospirenone / ethinyl estradiol	positive	neutral	neutral	neutral	neutral	positive
3909	Bisoprolol / hydrochlorothiazide	positive	neutral	neutral	neutral	positive	positive

- Un échantillon des prédictions pour la classe positive :

	Nom du médicament	Classe réelle	Random Forest	Régression Logistique	SVM	Classe Prédite (Moyenne)	Classe Prédite (Majorité)
0	Contrave	positive	neutral	neutral	neutral	neutral	positive
1	Topiramate	positive	negative	negative	positive	neutral	positive
2	Miconazole	positive	negative	negative	negative	negative	negative
3	Benzoyl peroxide / clindamycin	positive	positive	positive	positive	positive	positive
4	Belviq	positive	positive	positive	positive	positive	positive
5	Imitrex	positive	positive	positive	positive	positive	positive
6	Fingolimod	positive	positive	positive	positive	positive	positive
8	ParaGard	positive	negative	negative	negative	neutral	positive
9	Glatiramer	positive	positive	positive	positive	positive	positive
11	Paxil	positive	positive	positive	positive	positive	positive

- Un échantillon des prédictions sur la classe neutre:

	Nom du médicament	Classe réelle	Random Forest	Régression Logistique	SVM	Classe Prédite (Moyenne)	Classe Prédite (Majorité)
10	Buprenorphine / naloxone	neutral	positive	positive	positive	positive	positive
31	Levonorgestrel	neutral	neutral	neutral	neutral	positive	positive
37	Ethinyl estradiol / norgestrel	neutral	negative	negative	negative	neutral	positive
45	Etonogestrel	neutral	neutral	neutral	neutral	neutral	positive
61	Levonorgestrel	neutral	neutral	neutral	neutral	positive	positive
68	Lunesta	neutral	neutral	negative	negative	neutral	positive
75	Ethinyl estradiol / norethindrone	neutral	negative	negative	negative	neutral	positive
89	Liletta	neutral	neutral	neutral	neutral	neutral	positive
101	Lamictal	neutral	positive	positive	positive	positive	positive
127	Carbamazepine	neutral	positive	positive	positive	positive	positive

- Un échantillon des prédictions sur la classe négative:

	Nom du médicament	Classe réelle	Random Forest	Régression Logistique	SVM	Classe Prédite (Moyenne)	Classe Prédite (Majorité)
7	Ethinyl estradiol / levonorgestrel	negative	neutral	neutral	neutral	neutral	positive
22	Metronidazole	negative	neutral	neutral	neutral	neutral	positive
26	Deoxycholic acid	negative	positive	neutral	neutral	negative	negative
32	Viibryd	negative	negative	negative	negative	neutral	positive
33	Norethindrone	negative	negative	negative	negative	neutral	negative
35	Pramoxine	negative	positive	positive	positive	positive	positive
41	Zetia	negative	negative	negative	negative	neutral	positive
48	Etonogestrel	negative	neutral	neutral	neutral	neutral	positive
53	Amoxicillin	negative	neutral	neutral	neutral	neutral	positive
55	Depo-Provera	negative	negative	negative	negative	neutral	negative

Nous pouvons également afficher les cinq premiers médicaments de chaque classe pour chaque modèle :

```
Top 5 médicaments pour le modèle Random Forest:
  Classe positive: ['Benzoyl peroxide / clindamycin', 'Belviq', 'Imitrex', 'Fingolimod', 'Glatiramer']
  Classe negative: ['Viibryd', 'Norethindrone', 'Zetia', 'Depo-Provera', 'Rosuvastatin']


Top 5 médicaments pour le modèle Régression Logistique:
  Classe positive: ['Benzoyl peroxide / clindamycin', 'Belviq', 'Imitrex', 'Fingolimod', 'Glatiramer']
  Classe negative: ['Viibryd', 'Norethindrone', 'Zetia', 'Depo-Provera', 'Rosuvastatin']


Top 5 médicaments pour le modèle SVM:
  Classe positive: ['Topiramate', 'Benzoyl peroxide / clindamycin', 'Belviq', 'Imitrex', 'Fingolimod']
  Classe negative: ['Viibryd', 'Norethindrone', 'Zetia', 'Depo-Provera', 'Rosuvastatin']


Top 5 médicaments pour le modèle Classe Prédite (Moyenne):
  Classe positive: ['Benzoyl peroxide / clindamycin', 'Belviq', 'Imitrex', 'Fingolimod', 'Glatiramer']
  Classe negative: ['Deoxycholic acid', 'Miconazole', 'Afrin', 'Sensipar', 'Monistat 7']


Top 5 médicaments pour le modèle Classe Prédite (Majorité):
  Classe positive: ['Contrave', 'Topiramate', 'Benzoyl peroxide / clindamycin', 'Belviq', 'Imitrex']
  Classe negative: ['Deoxycholic acid', 'Norethindrone', 'Depo-Provera', 'Miconazole', 'Afrin']
```

Pour l'évaluation des performances des modèles de classification, plusieurs métriques sont utilisées :

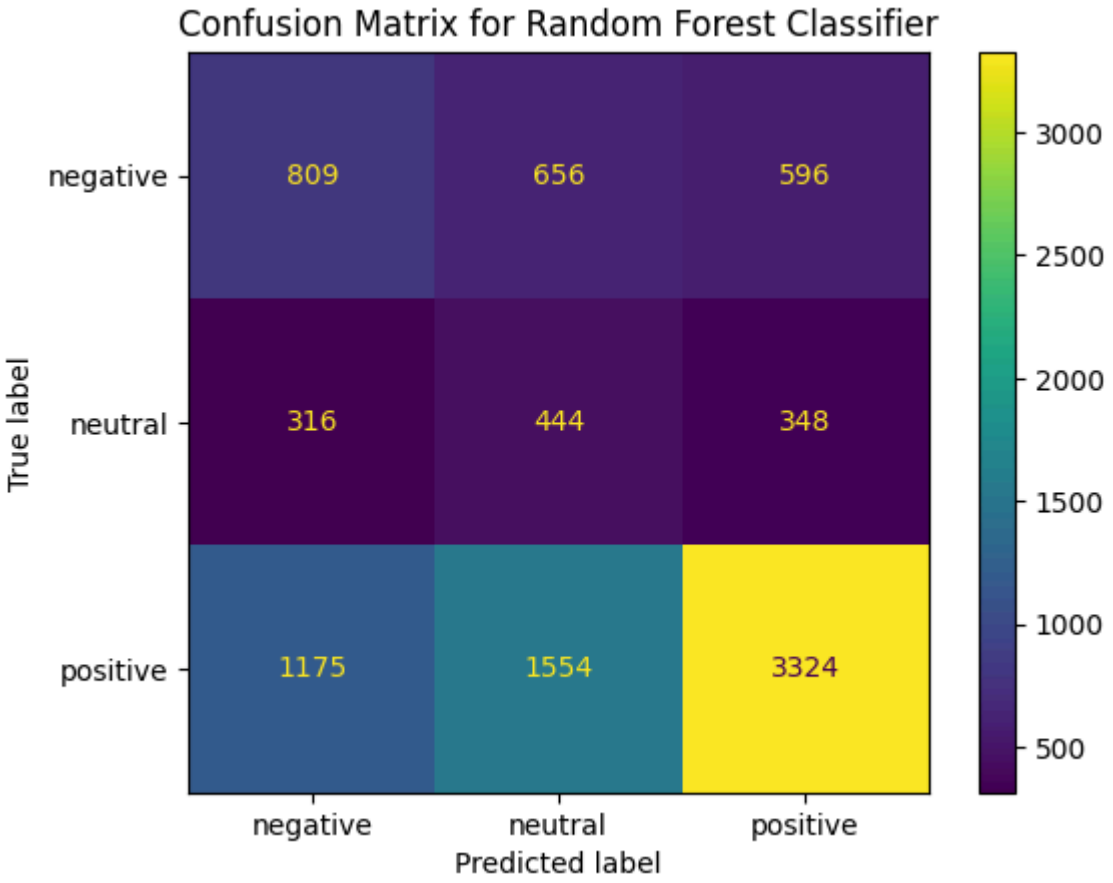
- **Précision (Precision)** : indique la proportion de prédictions positives qui sont correctes pour une classe.
- **Rappel (Recall)** : mesure la capacité du modèle à identifier toutes les instances d'une classe.
- **F1-score** : la moyenne harmonique de la précision et du rappel pour une classe.
- **Accuracy** : la proportion des prédictions correctes par rapport au total des prédictions.

Ces métriques peuvent être calculées grâce à la fonction `classification_report` de **scikit-learn**, que l'on peut compléter avec la matrice de confusion comparant les classes prédites aux classes réelles.

On obtient alors les rapports de performances suivants pour chaque modèles:

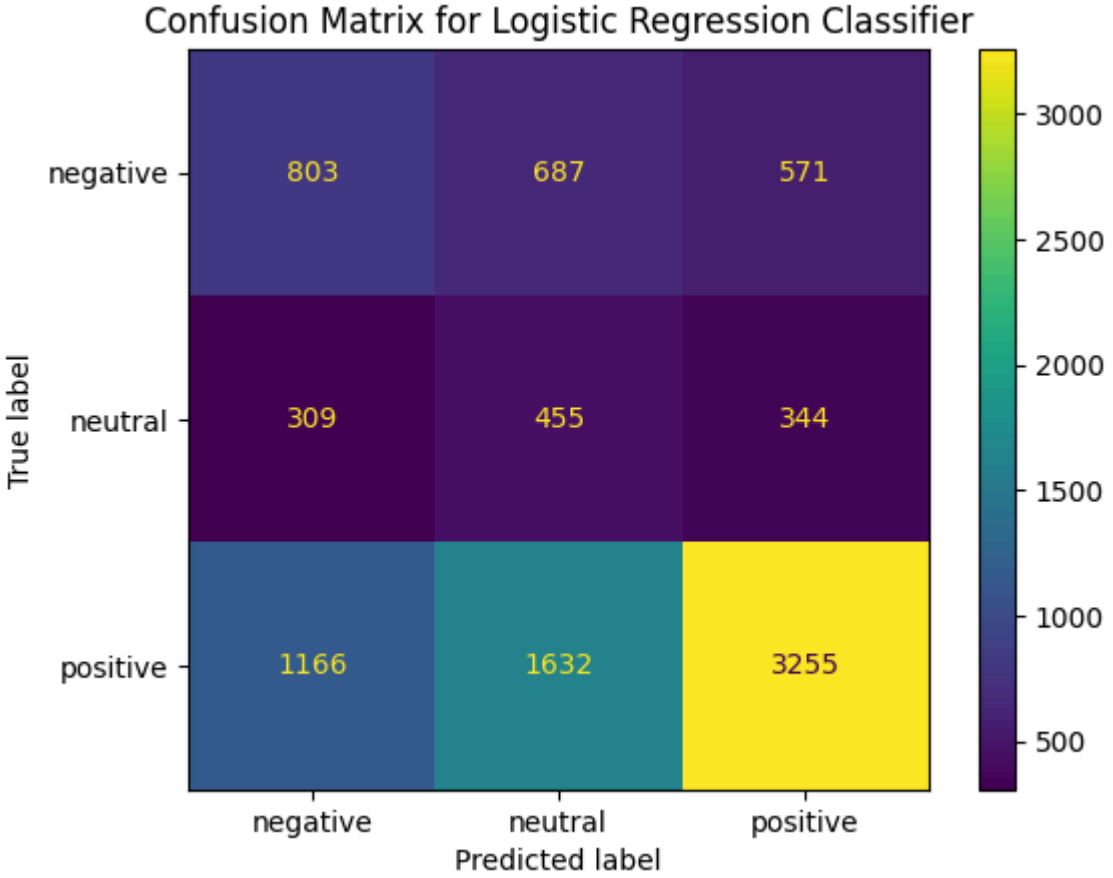
Random Forest Classifier:

	precision	recall	f1-score	support
negative	0.351739	0.392528	0.371016	2061.000000
neutral	0.167295	0.400722	0.236045	1108.000000
positive	0.778819	0.549149	0.644124	6053.000000
accuracy	0.496313	0.496313	0.496313	0.496313
macro avg	0.432618	0.447466	0.417061	9222.000000
weighted avg	0.609899	0.496313	0.534058	9222.000000



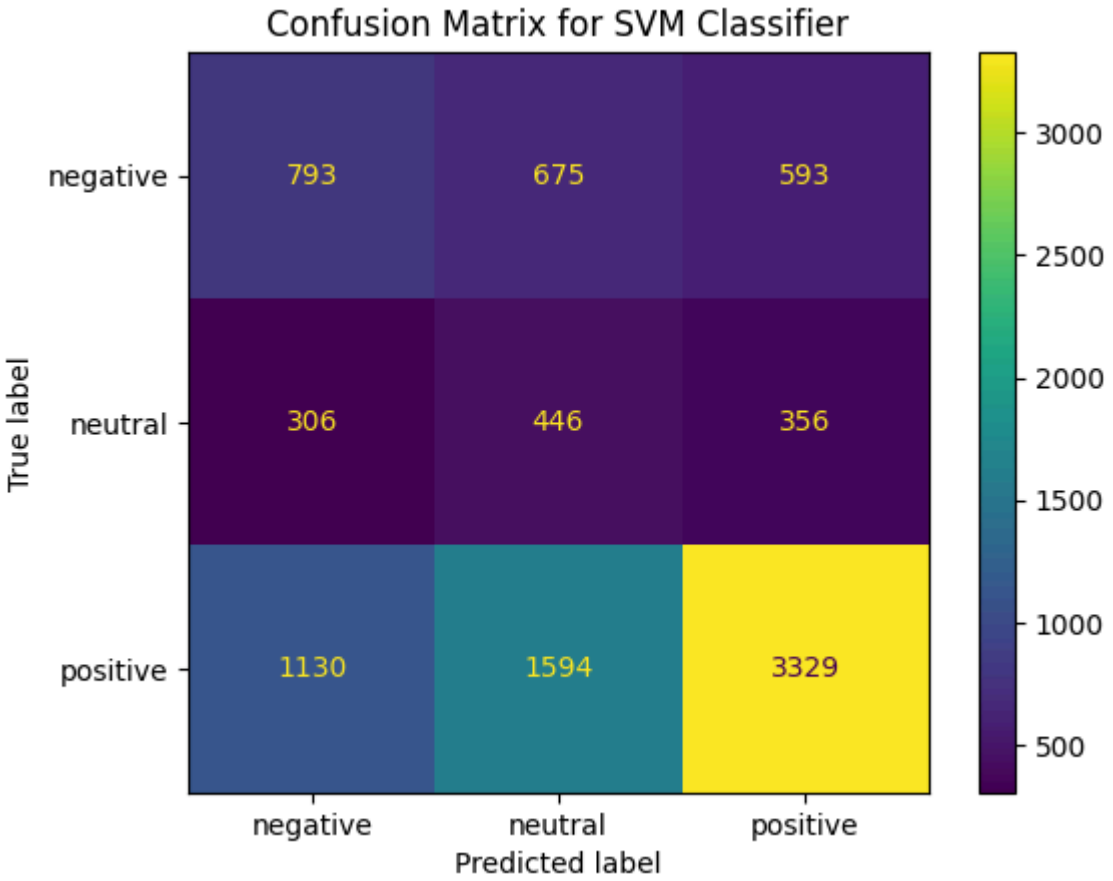
Logistic Regression Classifier:

	precision	recall	f1-score	support
negative	0.352502	0.389617	0.370131	2061.000000
neutral	0.164023	0.410650	0.234415	1108.000000
positive	0.780576	0.537750	0.636799	6053.000000
accuracy	0.489373	0.489373	0.489373	0.489373
macro avg	0.432367	0.446005	0.413782	9222.000000
weighted avg	0.610829	0.489373	0.528857	9222.000000



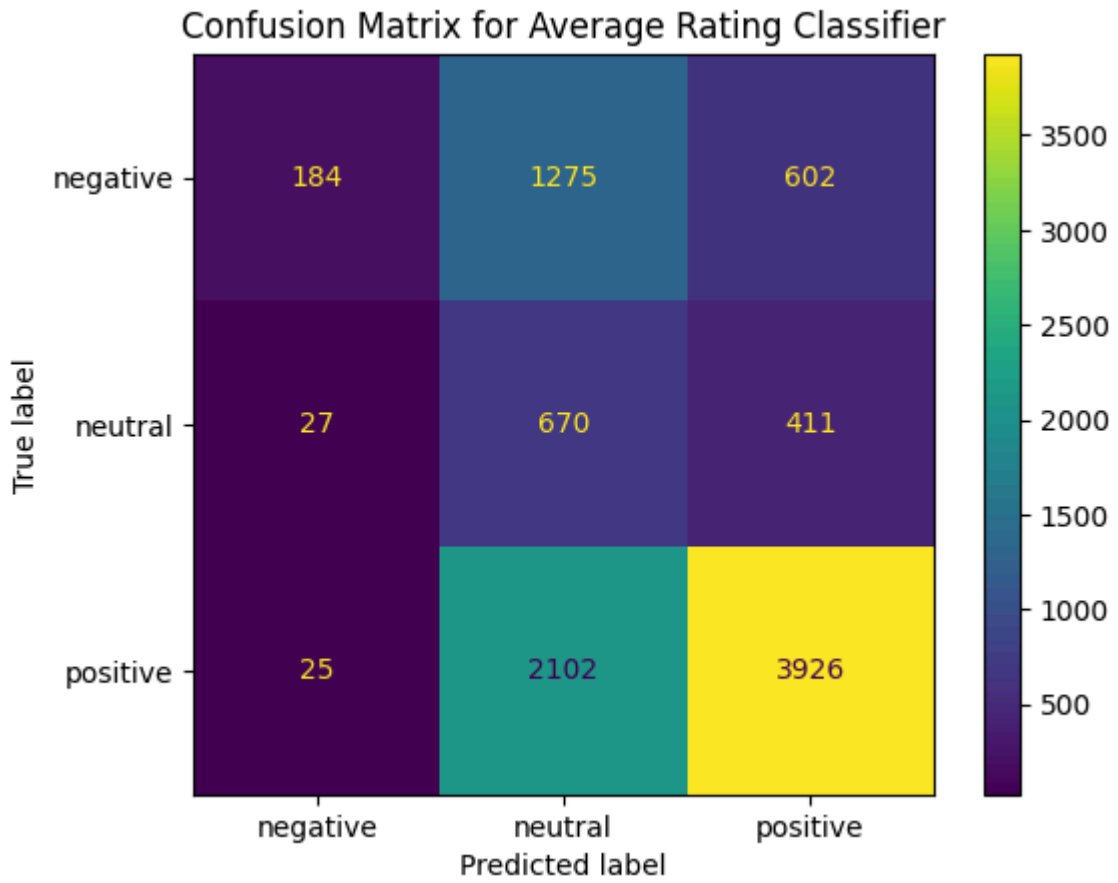
SVM Classifier:

	precision	recall	f1-score	support
negative	0.355765	0.384765	0.369697	2061.000000
neutral	0.164273	0.402527	0.233325	1108.000000
positive	0.778167	0.549975	0.644468	6053.000000
accuracy	0.495337	0.495337	0.495337	0.495337
macro avg	0.432735	0.445756	0.415830	9222.000000
weighted avg	0.610008	0.495337	0.533662	9222.000000



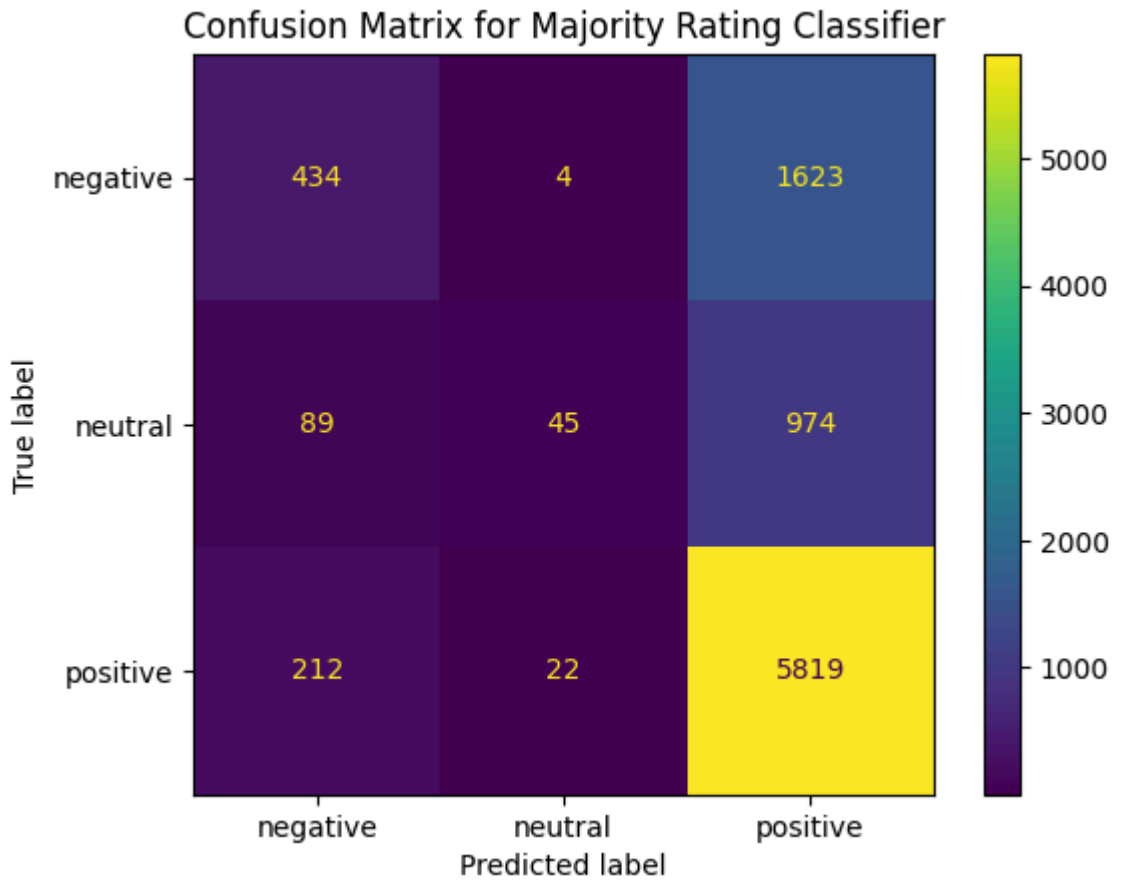
Average Rating Classifier:

	precision	recall	f1-score	support
negative	0.779661	0.089277	0.160209	2061.000000
neutral	0.165555	0.604693	0.259942	1108.000000
positive	0.794898	0.648604	0.714338	6053.000000
accuracy	0.518326	0.518326	0.518326	0.518326
macro avg	0.580038	0.447525	0.378163	9222.000000
weighted avg	0.715879	0.518326	0.535902	9222.000000



Majority Rating Classifier:

	precision	recall	f1-score	support
negative	0.590476	0.210577	0.310443	2061.000000
neutral	0.633803	0.040614	0.076336	1108.000000
positive	0.691421	0.961341	0.804340	6053.000000
accuracy	0.682932	0.682932	0.682932	0.682932
macro avg	0.638567	0.404178	0.397040	9222.000000
weighted avg	0.661939	0.682932	0.606493	9222.000000



Interprétation: En se basant sur la précision (accuracy) et le rappel (recall), on peut dire que la méthode de classification par classe majoritaire est la plus performante. Cependant, on remarque également qu'elle est fortement biaisée vers la classe "positive" et a beaucoup plus de difficultés à prédire les valeurs faibles.
La méthode de classification basée sur la moyenne des scores ou le modèle SVM peuvent donc représenter un choix intéressant pour équilibrer les rappels entre les classes, tout en maintenant une précision raisonnable.

Références

- Getting Started with Sentiment Analysis using Python
- A Survey on Text Classification: From Traditional to Deep Learning
- Classifying Drug Ratings Using User Reviews with Transformer-Based Language Models