Rapport sur les données

Prédiction de la qualité du vin

Présenté par Rolih DANGBO
Etudiant en Master 2 Big Data et Fouille de Données

Table des figures

Table des matières

1 - Objectif

L'objectif de notre travail consiste donc à prédire la qualité du vin en utilisant les algorithmes de machine learning avec PySpark en faisant du distribué. Dans nos jeux de données, la qualité du vin est représentée par une valeur quantitative qui varie entre 1 et 10 plus on s'approche de 1 plus le vin est moins bon, plus on s'approche de 10 plus le vin est très bon.

2 - Description des données

Les données du dataset ont été extraites à partir des caractéristiques d'un jeu de données de 4898 lignes de vins de couleur blanche. Ces données ont été pris sur le site de Kaggle (https://www.kaggle.com/sgus1318/winedata#winequality_white.csv). Nous avons pour ces données 12 colonnes constituées comme suit :

- fixed acidity : colonne qui décrit la teneur d'acidité fixe dans le vin.
- volatile acidity : colonne qui décrit la teneur d'acidité volatile dans le vin.
- citric acid : colonne qui décrit la teneur d'acide citrique contenu dans le vin.
- residual sugar : colonne qui décrit la teneur de sucre contenu dans le vin.
- chlorides : colonne qui décrit la teneur en chlorure de sodium contenu dans le vin.
- free sulfure dioxyde : colonne décrivant la teneur en dioxyde de souffre libre contenu dans le vin.
- total sulfure dioxyde : colonne décrivant la teneur en dioxyde de souffre total contenu dans le vin.
- density : colonne décrivant la densité du vin.
- pH : colonne décrivant le ph du vin.
- sulphates : colonne décrivant la teneur en sulphates de sodium contenu dans le vin.
- alcohol : colonne décrivant la teneur d'alcool contenu dans le vin.
- quality : colonne décrivant la qualité du vin. Cette colonne va servir comme label pour la prédiction de la qualité du vin (moins bon à très bon).

Toutes les données contenues dans les colonnes sont des données continues ou quantitatives.

3 - Problématique

La problématique de notre travail comme expliqué dans l'objectif consiste à prédire la qualité du vin (bon ou moins bon).

4 - Etat de l'art

Plusieurs algorithmes sont utilisés pour faire de la prédiction sur ces types de données parmi lesquels nous pouvons citer entre autres :

- La régression linéaire [1],
- Les arbres de décision [4],
- Les random forest [5],
- Les SVM ou encore [6],
- La régression logistique [2].

5 - Systèmes mis en place et résultats

a) Pour la prédiction de la qualité du vin voici les étapes suivies

- Exploration des données

fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
7.0	0.27	0.36	20.7	0.045	45.0	170.0	1.001	3.0	0.45	8.8	6
6.3	0.3	0.34	1.6	0.049	14.0	132.0	0.994	3.3	0.49	9.5	6
8.1	0.28	0.4	6.9	0.05	30.0	97.0	0.9951	3.26	0.44	10.1	6
7.2	0.23	0.32	8.5	0.058	47.0	186.0	0.9956	3.19	0.4	9.9	6
7.2	0.23	0.32	8.5	0.058	47.0	186.0	0.9956	3.19	0.4	9.9	6
8.1	0.28	0.4	6.9	0.05	30.0	97.0	0.9951	3.26	0.44	10.1	6
6.2	0.32	0.16	7.0	0.045	30.0	136.0	0.9949	3.18	0.47	9.6	6
7.0	0.27	0.36	20.7	0.045	45.0	170.0	1.001	3.0	0.45	8.8	6
6.3	0.3	0.34	1.6	0.049	14.0	132.0	0.994	3.3	0.49	9.5	6
8.1	0.22	0.43	1.5	0.044	28.0	129.0	0.9938	3.22	0.45	11.0	6
8.1	0.27	0.41	1.45	0.033	11.0	63.0	0.9908	2.99	0.56	12.0	5
8.6	0.23	0.4	4.2	0.035	17.0	109.0	0.9947	3.14	0.53	9.7	5
7.9	0.18	0.37	1.2	0.04	16.0	75.0	0.992	3.18	0.63	10.8	5
6.6	0.16	0.4	1.5	0.044	48.0	143.0	0.9912	3.54	0.52	12.4	7
8.3	0.42	0.62	19.25	0.04	41.0	172.0	1.0002	2.98	0.67	9.7	5
6.6	0.17	0.38	1.5	0.032	28.0	112.0	0.9914	3.25	0.55	11.4	7
6.3	0.48	0.04	1.1	0.046	30.0	99.0	0.9928	3.24	0.36	9.6	6
6.2	0.66	0.48	1.2	0.029	29.0	75.0	0.9892	3.33	0.39	12.8	8

Figure 1 : Exploration de notre jeu de données

- Exploration des données (suite)

	0	1	2	3	4
summary	count	mean	stddev	min	max
fixed acidity	4898	6.854787668436075	0.8438682276875127	3.8	14.2
volatile acidity	4898	0.27824111882401087	0.10079454842486532	0.08	1.1
citric acid	4898	0.33419150673743736	0.12101980420298254	0.0	1.66
residual sugar	4898	6.391414863209486	5.072057784014878	0.6	65.8
chlorides	4898	0.0457723560636995	0.021847968093728805	0.009	0.346
free sulfur dioxide	4898	35.30808493262556	17.00713732523259	2.0	289.0
total sulfur dioxide	4898	138.36065741118824	42.498064554142985	9.0	440.0
density	4898	0.9940273764801896	0.002990906916936997	0.98711	1.03898
pH	4898	3.1882666394446693	0.15100059961506673	2.72	3.82
sulphates	4898	0.4898468762760325	0.11412583394883222	0.22	1.08
alcohol	4898	10.514267047770149	1.230620567752269	8.0	14.2
quality	4898	5.87790935075541	0.8856385749678322	3	9

Figure 2 : Statistique descriptive de notre jeu de données

- Etude de la corrélation entre les données et sélection des variables pertinentes

▸ (1) Spark Jobs

feature:	corcoef	cc_log	covar
=======	=======	======	=====
alcohol:	0.445	0.404	0.424
density:	-0.404	-0.372	-0.001
residual sugar:	-0.278	-0.255	-1.316
pH:	0.227	0.203	0.030
total sulfur dioxide:	-0.193	-0.169	-7.554
volatile acidity:	-0.181	-0.209	-0.017
chlorides:	-0.172	-0.157	-0.003
sulphates:	0.103	0.105	0.009
free sulfur dioxide:	-0.084	-0.062	-1.263
citric acid:	0.022	0.027	0.002

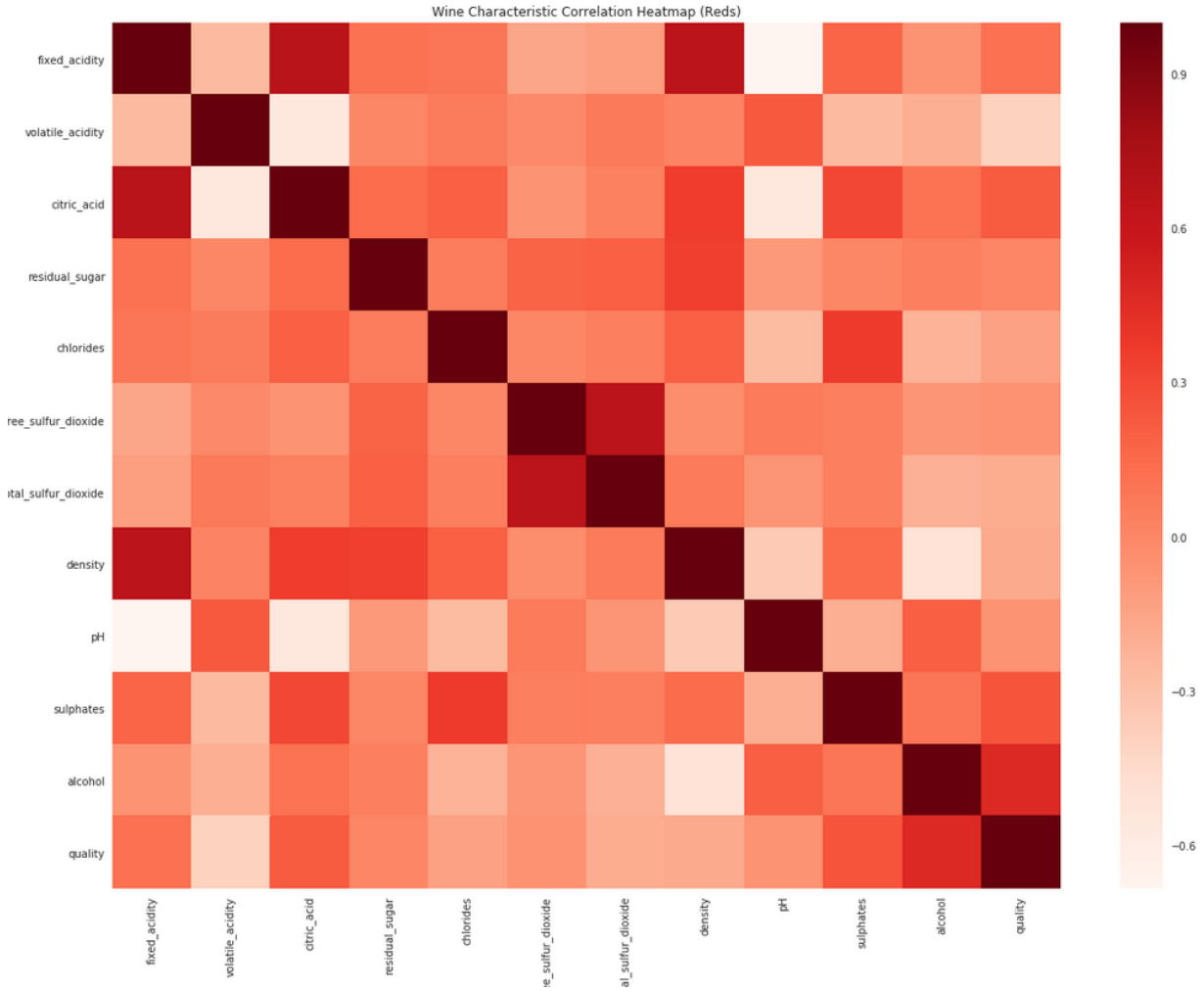

Figure 3 : Etude de la corrélation entre les variables

Nous avons donc décider de ne garder que des variables les plus importantes pour la prédiction (variables d'entré : "citric acid","pH","sulphates","alcohol" et variables de sorties : "quality").

- Prédiction avec la régression linéaire

Regression Linéaire

```
from pyspark.ml.regression import LinearRegression
from pyspark.mllib.classification import LogisticRegressionWithLBFGS
from pyspark.mllib.util import MLUtils
from pyspark.mllib.evaluation import MulticlassMetrics
lr = LinearRegression(featuresCol = "features", labelCol="quality", maxIter=10, regParam=0.3, elasticNetParam=0.8)
lr_model = lr.fit(train_df)
prediction = lr_model.transform(test_df)
print("Coefficients: " + str(lr_model.coefficients))
print("Intercept: " + str(lr_model.intercept))
```

Figure 4 : Application de la régression linéaire pour prédire la qualité du vin

features	quality	prediction
[0.0,3.1,0.4,10.9]	4	5.919622407423278
[0.0,3.27,0.67,10.8]	6	5.908678155906089
[0.0,3.3,0.63,9.9]	6	5.8101798922513845
[0.0,3.31,0.38,9.4]	4	5.7554586346654375
[0.0,3.32,0.51,10.5]	4	5.875845401354521
[0.0,3.35,0.61,9.9]	5	5.8101798922513845
[0.0,3.37,0.38,11.2]	4	5.9524551619748465
[0.0,3.63,0.4,9.7]	4	5.788291389217005
[0.01,3.24,0.35,9.5]	5	5.766402886182627
[0.01,3.38,0.36,1...]	7	6.072841928663929
[0.02,3.14,0.47,9.8]	4	5.799235640734195
[0.02,3.2,0.32,9.0]	5	5.71168162859668
[0.03,3.34,0.38,9.2]	5	5.733570131631058
[0.04,3.14,0.4,11.0]	4	5.930566658940467
[0.04,3.17,0.39,1...]	4	5.843012646802952
[0.04,3.22,0.51,1...]	5	5.843012646802952
[0.04,3.24,0.36,9.6]	6	5.777347137699817
[0.04,3.24,0.61,9.9]	6	5.8101798922513845
[0.04,3.26,0.54,9.2]	5	5.733570131631058
[0.04,3.36,0.33,1...]	8	6.050953425629551

```
Model accuracy: 85.061%
MSE: 72.354%
MAE: 64.263%
r2: 10.780%
numIterations: 6
objectiveHistory: [0.5, 0.49720004724685996, 0.4875182034851134, 0.48751543148877613, 0.48751543069512965, 0.48751543069490366]
+--------------------+
|           residuals|
+--------------------+
| 0.10226609561110056|
| -1.7335701316310583|
| -0.6897931255623018|
| -1.0837861801811188|
| 0.09132184409391098|
|-0.08378618018111883|
| 0.18982010774861546|
|-0.08378618018111883|
|-0.23700570142177035|
| -0.9196224074232777|
| -0.8430126468029524|
| -1.0290649225951718|
```

Figure 5 : Evaluation du modèle pour la régression linéaire

- Prédiction avec le random forest

Random Forest

```
from pyspark.mllib.tree import RandomForest
from time import *

start_time = time()

model = RandomForest.trainClassifier(train_df, numClasses=15, categoricalFeaturesInfo={}, \
    numTrees=20, featureSubsetStrategy="auto", impurity="gini", \
    maxDepth=20,seed=RANDOM_SEED)
#RF_MAX_DEPTH
#RF_NUM_TREES
end_time = time()
elapsed_time = end_time - start_time
print("Time to train model: %.3f seconds" % elapsed_time)
```

Figure 6 : Application du Random Forest pour prédire la qualité du vin

```
Model accuracy: 64.566%
```

Figure 7 : Evaluation du modèle pour le Random Forest

Au vue de l'utilisation de ces deux algorithmes nous pouvons en conclure que nous avons un meilleur résultat avec la régression linéaire (85% contre 64% pour le random forest).

6 - Conclusion

Ce travail consistait donc à prédire la qualité et la couleur du vin. Il nous a permis de monter encore plus en compétences sur les différents algorithmes de machine learning tel que la régression linéaire, le random forest et le svm.

Comme perspectives nous prévoyons de tester également d'autres algorithmes tels que la régression logistique et les arbres de décision afin de comparer avec les résultats obtenus ci-dessous.

7 - Références

- [1]. Christophe Chesneau. Modèles de régression. Master. France. 2015.
- [2]. Maria Koutina, Katia Kermanidis. Predicting Postgraduate Students' Performance Using Machine Learning Techniques. 12th Engineering Applications of Neural Networks (EANN 2011) and 7thArtificial Intelligence Applications and Innovations (AIAI), Sep 2011, Corfu, Greece.
- [3]. Dana Marinca, Pascale Minet, Nesrine Ben Hassine. An efficient learning technique to predict linkquality in WSN. PIMRC 2014 - 25th Annual International Symposium on Personal, Indoor and Mobile Radio Communications, Sep 2014, Washington, United States.
- [4]. Gilbert Ritschard, Simon Marcellin, Djamel A. Zighed, arbre de décision pour données déséquilibrées : sur la complémentarité de l'intensité d'implication et de l'entropie décentrée, Département d'économétrie, Université de Genève, Laboratoire ERIC, Université de Lyon 2.
- [5]. Audrey Poterie. Arbres de décision et forêts aléatoires pour variables groupées. Statistiques [math.ST]. INSA de Rennes, 2018.
- [6]. Fabien Lauer, Gérard Bloch. Méthodes SVM pour l'identification. Journées Identification et Modélisation Expérimentale (JIME'2006), Nov 2006, Poitiers, France.