Fruit Price

Parmi les trois études, j'ai choisi Fruit Price. Au début, je voulais choisir NLP, mais comme c'est un sujet de Deep Learning, je ne l'ai pas choisi. Parce que je n'ai qu'un ordinateur portable avec des performances insuffisantes, il peut être difficile d'entraîner un modèle de RNN pour la classification.

# Charger des données

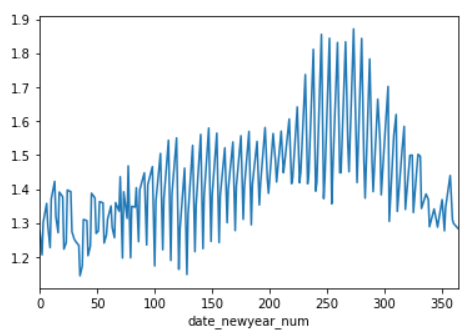
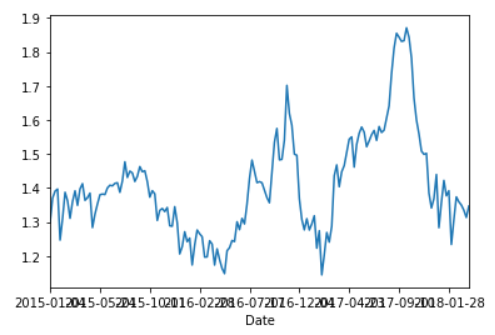
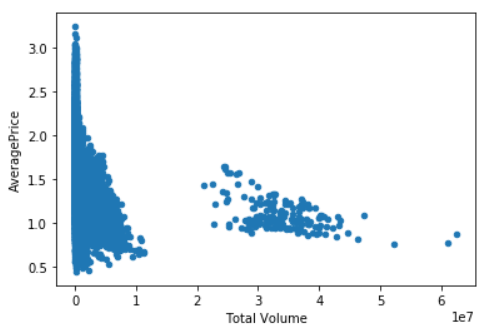
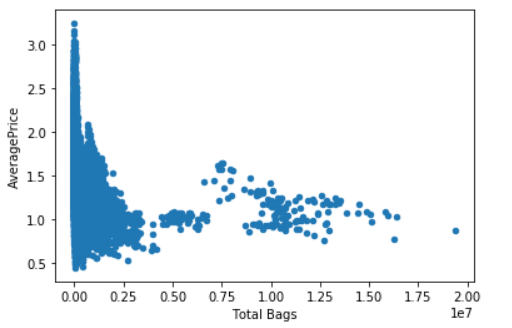
La première chose est de charger les données. J'ai utilisé Pandas pour lire votre fruit\_price.csv dans un dataframe appelé « df ». Il contient les colonnes suivantes:

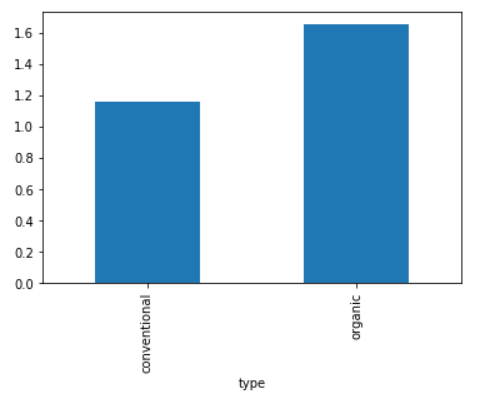
* Unnamed: 0 : Il s'agit du numéro d'index des données de chaque région, je n'en ai pas besoin, je peux donc le supprimer
* Date : Ce sont des données très intéressantes, je peux les utiliser, mais je dois d’abord les traiter.
* AveragePrice : C'est sont les prix que j'ai besoin de prédire.
* Total Volume, 4046, 4225, 4770, Total Bags,Small Bags,Large Bags,XLarge Bags: Des facteurs qu'on peut utiliser.
* type,region: ces sont des variables qualitatives, je peux faire One Hot Encoding
* year: year est déjà incluse dans Date, je peux faire One Hot Encoding aussi

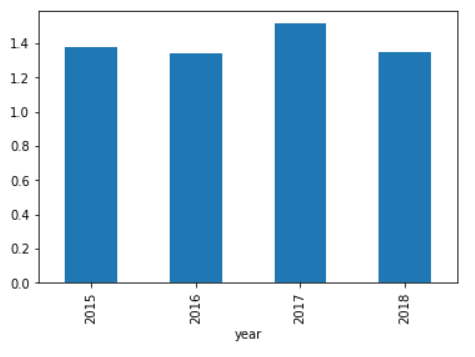
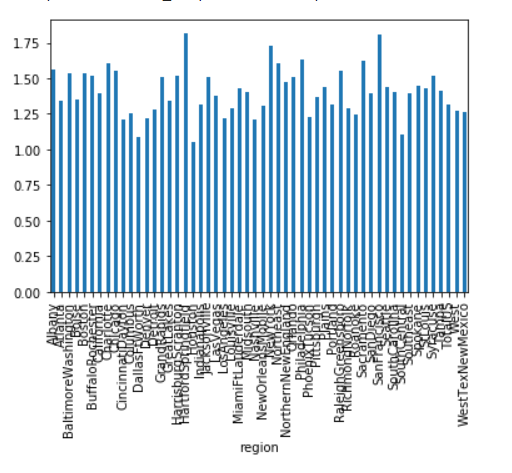
Après cela, j'ai utilisé « describe() » pour observer les statistiques des données. J’ai trouvé que "Date" était un facteur très utile, mais je ne pourrais pas l'utiliser directement. Puisque nous avons des données de 2015-2018, j'ai converti "Date" en « combien de jours se sont écoulés depuis le nouvel an de cette année ». Cela transformera "Date" en une variable quantitative. Ex : 2015-12-27 -> 360, 2018-01-07 -> 6. Cette nouvelle colonne s'appelle 'date\_newyear\_num'

# Data Visualisation

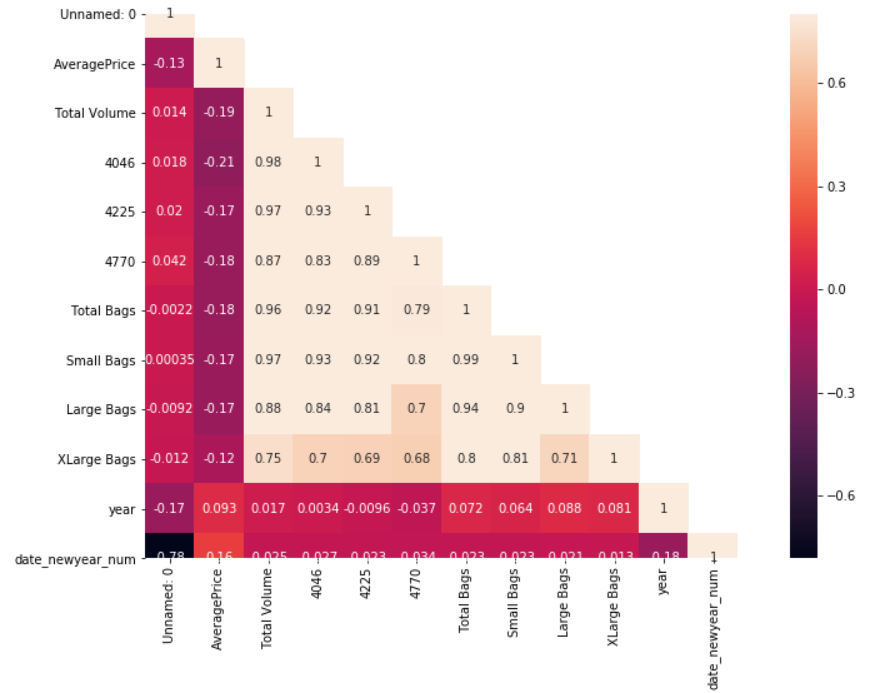
Tout d'abord, j’ai fait Data Visualisation pour comprendre la relation entre chaque facteur et le prix. J’ai fait plusieurs figures :







J’ai fait aussi un heatmap pour présenter les corrélations entre différents facteurs.



J'ai trouvé que la plupart des facteurs sont liés à "prix", mais ils ne sont pas très pertinents. Évidemment, je dois faire des traitements sur les variables.

# Nettoyage de données

## Supprimer les colonnes inutiles

D'abord, j’ai utilisé la fonction de « count() » pour déterminer s'il existait des valeurs manquantes et les résultats montraient qu'il n'y avait pas de valeurs manquantes dans les données.

Et puis, Il ressemblait que " volume total" = "4046" + "4225" + "4770" + "Total Bags". Cependant, je devais le vérifier. J'ai donc construit un nouveau dataframe "re\_Total\_Volume", qui stocke la différence entre "4046" + "4225" + "4770" + "Total Bags" et "Total Volume", j’ai trouvé qu'il y a pas mal de différences entre les deux. Comme je ne savais pas comment le "Total Volume " est calculé, j'ai décidé de le garder.

Par contre, j’ai fait la même chose pour «Total Bags» et "Small Bags"+"Large Bags"+"XLarge Bags". Mais j’ai trouvé que la valeur absolue de la différence entre "Total Bags" et "Small Bags + Large Bags + XLarge Bags" est toujours inférieure à 1. Cela montrait que "Total Bags" était approximativement égal à "Small Bags + Large Bags + XLarge Bags", j’ai pensé donc que la colonne "Total Bags" pouvait être supprimée car les informations qu'elle contenait étaient bien exprimées dans les trois autres colonnes.

Ensuite, pour les trois variables qualitatives "type", "année" et "région", j'ai utilisé One Hot Encoding pour les traiter.

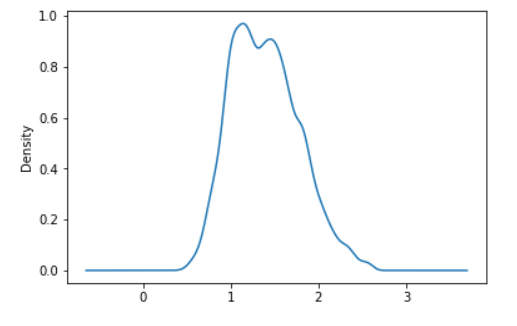
Finalement, j'ai supprimé 'Unnamed: 0', 'Date', 'Total Bags' trois colonnes.

## Les Valeurs Aberrantes

Puisqu'il n'y avait pas de valeurs nulles dans les données, j’ai appliqué directement la méthode des trois écarts-types pour supprimer les Valeurs Aberrantes (Outliers) pour "AveragePrice". Finalement, j’ai supprimé 131 valeurs aberrantes.

# Entraîner les modèles Machine Learning

Premièrement, «prédiction des prix» s'agit d'un problème de régression, donc si la valeur cible suit la loi normale, cela fonctionnera bien pour de nombreux modèles. J’ai fait un plot KDE pour "AveragePrice".



Cette figure n'est pas une bonne distribution de la loi normale, il peut donc être difficile d'utiliser certains types de modèle linéaire. Au final, j'ai choisi deux modèles plus couramment utilisés : Random Forest Regression et Gradient Boosting Regression. Parce que ces deux modèles fonctionnent généralement très bien et ne sont pas les modèles linéaires habituels.

J’ai divisé au hasard le dataset en trainset et testset, le rapport 7: 3. Pour évaluer les modèles, j’ai utilisé RMSLE (root mean squared logarithmic error) pendant l’entraînement des modèles, mais j’ai aussi utilisé RMSE (root mean square error) pour les résultats finals. Les deux indicateurs sont capables d'évaluer la différence entre la prédiction et la valeur réelle. Ne que ce soit RMSE ou RMSLE, plus les deux sont petits, le modèle est meilleur.

## Machine Learning

Pour l’entraînement des deux modèles, j'ai utilisé la méthode Grid Search pour optimiser les paramètres. Cependant, en raison des performances insuffisantes de mon ordinateur, je n'ai pas utilisé beaucoup de paramètres. Pour des paramètres spécifiques, veuillez-vous référer à mon code. Ensuite, j'ai utilisé les paramètres optimaux et trainset pour entraîner les modèles, et puis, utilisé les modèles pour la prédiction de testset, puis j'ai comparé la prédiction avec les données réelles pour évaluer mes modèles en utilisant RMSLE et RMSE.

### Random Forest Regressor

Les paramètres que j’ai utilisé sont : {'min\_samples\_leaf': 1, 'min\_samples\_split': 2, 'n\_estimators': 2000, 'n\_jobs': -1, 'random\_state': 0}

Les résultats d’évaluation :

RMSE: 0.1188

RMSLE: 0.0487

Gradient Boosting Regressor

Les paramètres que j’ai utilisé sont : {'learning\_rate': 0.1, 'max\_depth': 4, 'min\_samples\_leaf': 3, 'n\_estimators': 1000}

Les résultats d’évaluation:

RMSE: 0.1142

RMSLE: 0.0463

# Les résultats

D'après les résultats ci-dessus, j'ai vu que mon meilleur modèle est "Gradient Boosting Regression", RMSE est de 0,1142 et RMSE est de 0,0463, ce qui est un bon résultat. Après l’entraînement, j'ai rechargé mon modèle et fait un classement de l'influence de divers facteurs sur le prix.

Les dix premier facteurs sont :

* {'type\_conventional': 0.2098536258268962}
* {'type\_organic': 0.17364420040853612}
* {'date\_newyear\_num': 0.09219699701717347}
* {'Large Bags': 0.08839090751951344}
* {'4046': 0.07998609099786888}
* {'4225': 0.053404895548166416}
* {'year\_2017': 0.04950199964508371}
* {'Total Volume': 0.04077686667182434}
* {'Small Bags': 0.03767649611934359}
* {'4770': 0.020826132298218076}

Par conséquent, le plus important facteur est "type", si les fruits sont «conventional» ou «organic», ce qui est évident. L'importance de «date\_newyear\_num» est de 0,0921, ce qui indique que les différentes périodes de l'année ont un certain impact sur le prix des fruits.

# Réutilisation des modèles

J’ai sauvegardé le modèle au format pkl, je peux le recharger et utiliser facilement. Mais parce que j'ai fait des traitements sur dataset, si quelqu'un veut utiliser mon modèle, il doit y avoir ces colonnes:

'AveragePrice', 'Total Volume', '4046', '4225', '4770', 'Small Bags', 'Large Bags', 'XLarge Bags', 'date\_newyear\_num', 'type\_conventional', 'type\_organic', 'year\_2015', 'year\_2016', 'year\_2017', 'year\_2018', 'region\_Albany', 'region\_Atlanta', 'region\_BaltimoreWashington', 'region\_Boise', 'region\_Boston', 'region\_BuffaloRochester', 'region\_California', 'region\_Charlotte', 'region\_Chicago', 'region\_CincinnatiDayton', 'region\_Columbus', 'region\_DallasFtWorth', 'region\_Denver', 'region\_Detroit', 'region\_GrandRapids', 'region\_GreatLakes', 'region\_HarrisburgScranton', 'region\_HartfordSpringfield', 'region\_Houston', 'region\_Indianapolis', 'region\_Jacksonville', 'region\_LasVegas', 'region\_LosAngeles', 'region\_Louisville', 'region\_MiamiFtLauderdale', 'region\_Midsouth', 'region\_Nashville', 'region\_NewOrleansMobile', 'region\_NewYork', 'region\_Northeast', 'region\_NorthernNewEngland', 'region\_Orlando', 'region\_Philadelphia', 'region\_PhoenixTucson', 'region\_Pittsburgh', 'region\_Plains', 'region\_Portland', 'region\_RaleighGreensboro', 'region\_RichmondNorfolk', 'region\_Roanoke', 'region\_Sacramento', 'region\_SanDiego', 'region\_SanFrancisco', 'region\_Seattle', 'region\_SouthCarolina', 'region\_SouthCentral', 'region\_Southeast', 'region\_Spokane', 'region\_StLouis', 'region\_Syracuse', 'region\_Tampa', 'region\_TotalUS', 'region\_West', 'region\_WestTexNewMexico'