Online News Popularity

Python project

by Clémence Delouche, Guillaume Doria & Léa Feldman (DIA3)

Summary

Context
The OnlineNews.csv Dataset

We used the Online News Popularity dataset provided by the UCI Machine Learning Repository. This dataset is part of a research project aiming at predicting the popularity of an article on one hand and optimizing the features to improve it on the other hand. The interesting part is that we analyze article prior to their publication to help writers have a better understanding of their work before putting it out to the world. The articles were published on Mashable (*www.mashable.com*).

If you want to download it, follow this link :
https://archive.ics.uci.edu/ml/datasets/Online+News+Popularity#

Dataset
key figures

- 61 attributes
- 39644 instances
- The dataset area is business

58 PREDICTIVES ATTRIBUTES

- Number of link in the artcile
- The day when it's published
- Number of worlds in the title
- etc

2 NON-PREDICTIVES ATTRIBUTES

- URL
- Timedelta (days between the article publication and the dataset acquisition)

1 GOAL FIELD

- The number of shares (popularity)

Project
Development

1

PROBLEMATICS AND SOLUTIONS

2

DATA VISUALISATION

3

DATA PRE-PROCESSING

4

MODELLING

5

API

Problematics & Solutions

How can we predict the popularity ?

We define popularity by number of shares

&

We create a ML model to run the estimation

How to know which data or variable is useful ?

Which variables are most correlated to the share parameter ?

Correlation matrix

Histograms

(to split popular and unpopular depending on the number of days and channels)

Variables we created

Publish Day

In [11]:

```
df['publish_day']= 0 #Creation of a new column 'publish_day'
df['publish_day'] = np.where(df[' weekday_is_monday']==1, 'Monday',df['publish_day'])
df['publish_day'] = np.where(df[' weekday_is_tuesday']==1, 'Tuesday',df['publish_day'])
df['publish_day'] = np.where(df[' weekday_is_wednesday']==1, 'Wednesday',df['publish_day'])
df['publish_day'] = np.where(df[' weekday_is_thursday']==1, 'Thursday',df['publish_day'])
df['publish_day'] = np.where(df[' weekday_is_friday']==1, 'Friday',df['publish_day'])
df['publish_day'] = np.where(df[' weekday_is_saturday']==1, 'Saturday',df['publish_day'])
df['publish_day'] = np.where(df[' weekday_is_sunday']==1, 'Sunday',df['publish_day'])
```

Channel

In [13]:

```
df['channel'] = 0 #Creation of a new column "channel"
df['channel'] = np.where(df[' data_channel_is_lifestyle']==1, 'Lifestyle',df['channel'])
df['channel'] = np.where(df[' data_channel_is_entertainment']==1, 'Entertainement',df['channel']
df['channel'] = np.where(df[' data_channel_is_bus']==1, 'Business',df['channel'])
df['channel'] = np.where(df[' data_channel_is_socmed']==1, 'Social media',df['channel'])
df['channel'] = np.where(df[' data_channel_is_tech']==1, 'Tech',df['channel'])
df['channel'] = np.where(df[' data_channel_is_world']==1, 'World',df['channel'])
df
```

These variables were created to facilitate the understanding of the data

Popularity

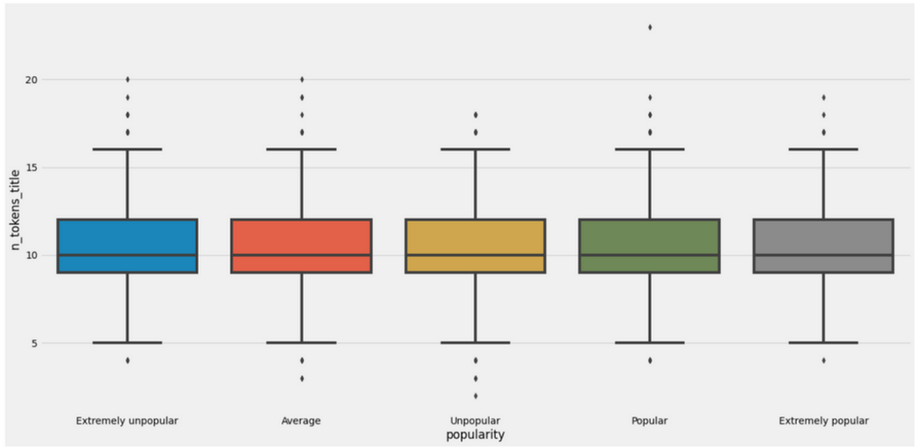
In [16]:

```
df['popularity'] = df[' shares'] #Creation of a new column in the dataset
df['popularity'] = df['popularity'].apply(lambda x: 'Extremely popular' if x > 10800 else 'Popular' if x > 2300  else 'Average' if x > 1400 else 'Ur
df
```

We created this variable to have a scale of popularity to make fixing outliers easier

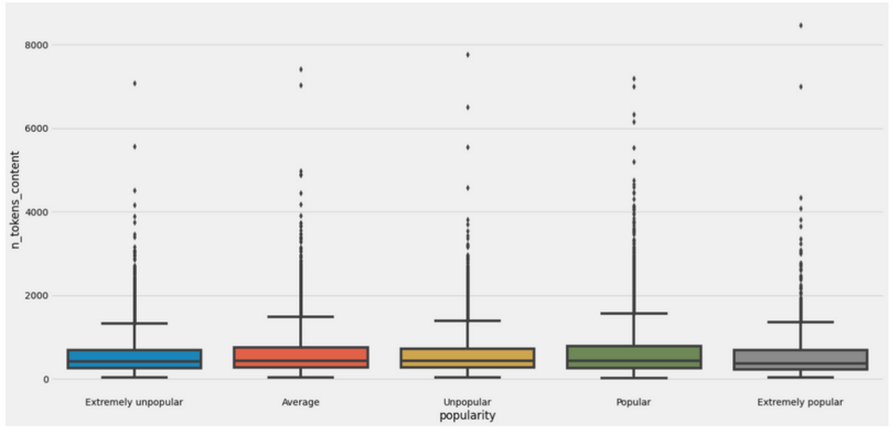
Fixing Outliers

Out[17]:



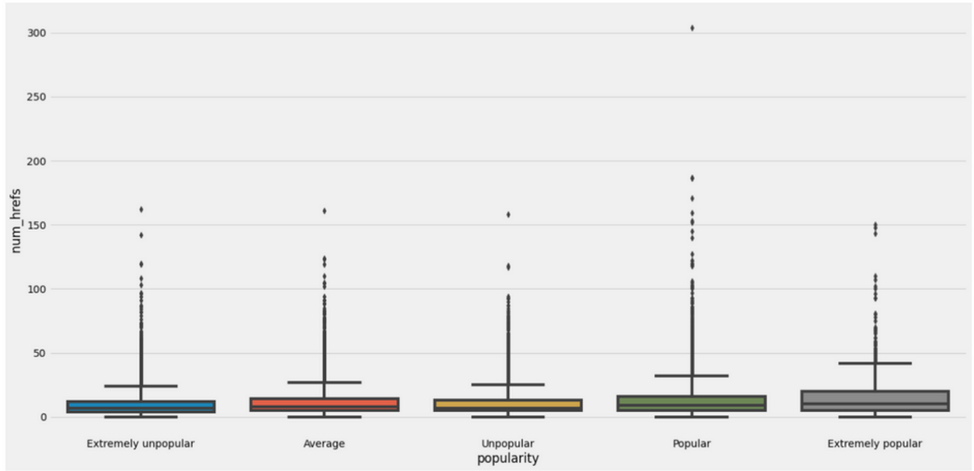
Popularity by number of worlds in the title

Out[21]:



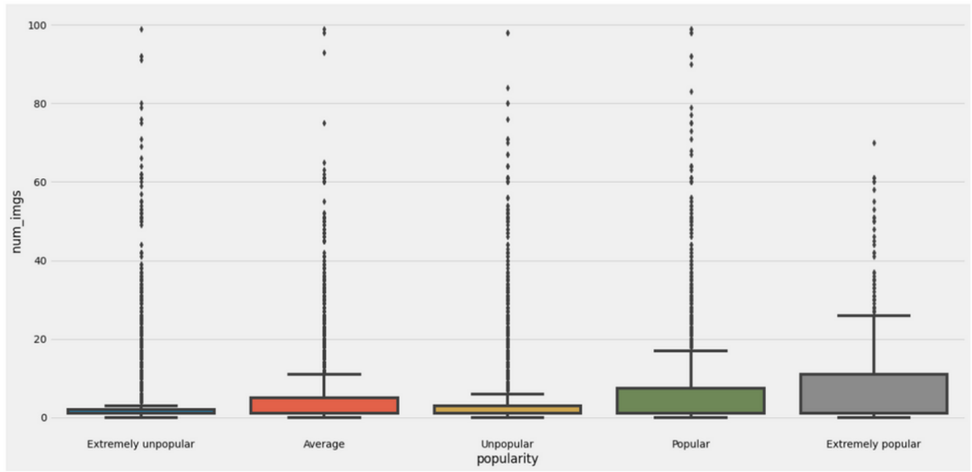
Popularity by number of worlds in the article

Out[23]:



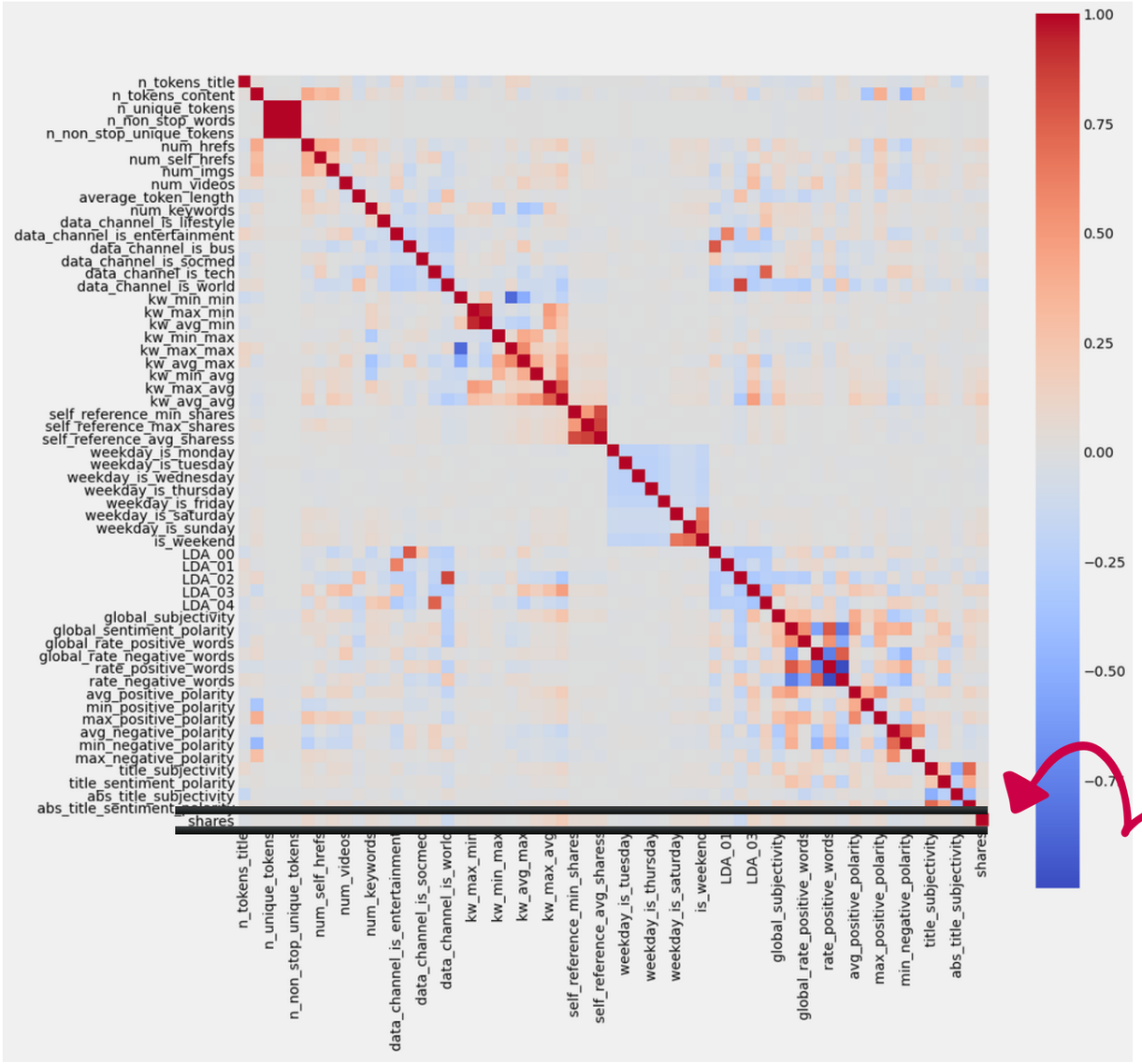
Popularity by number of links

Out[25]:



Popularity by number of images

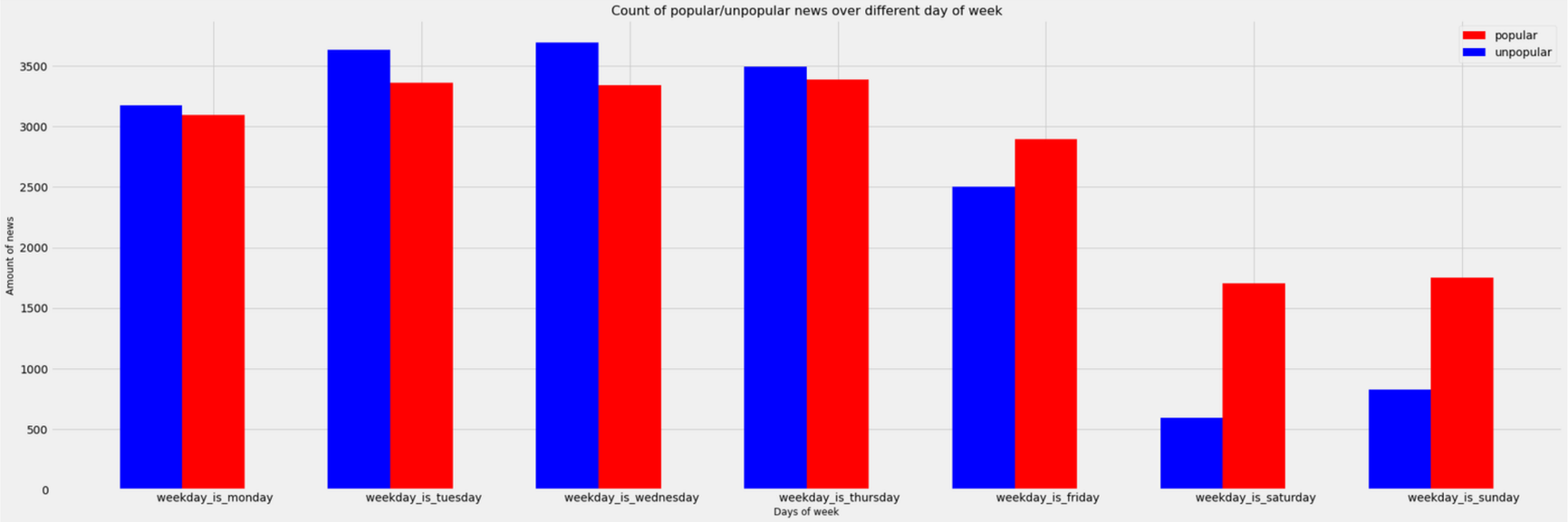
Data Visualisation

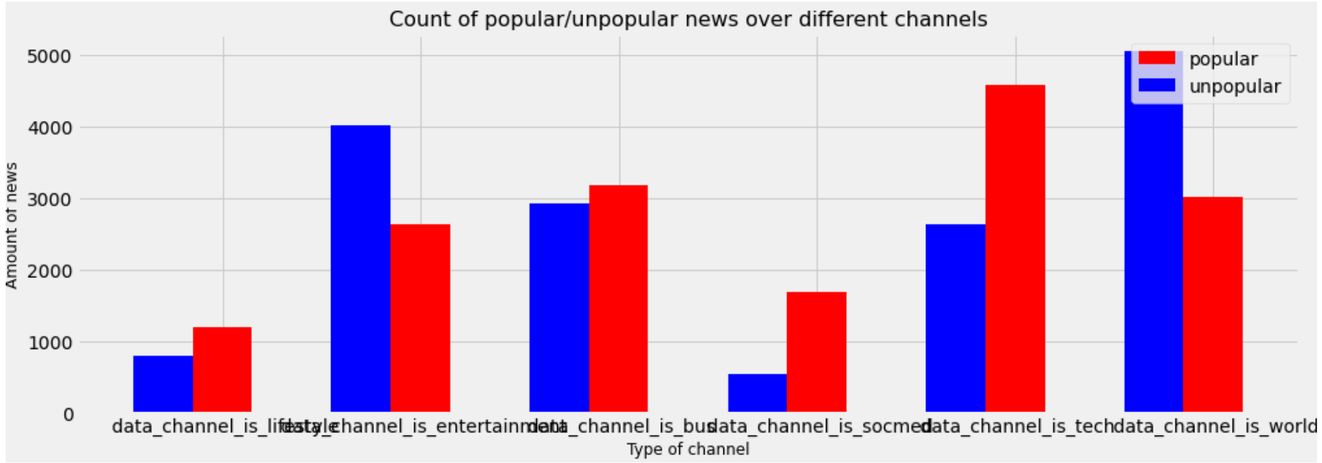


Our data is not well correlated with the "shares" parameter

Correlation Matrix

Data Visualisation





Modelling

We tried different classification method to find the best model for our Machine Learning : **Linear Regression, Decision Tree, Random Forest.**

The model with the most important chance of success is

RANDOM FOREST

Before modelling
train the data

Train-Test Split

We need to split our data to a train set (80%) and test set (20%). We are trying to predict the variable 'shares'.

```
(29976, 18)
(7494, 18)
(29976,)
(7494,)
```

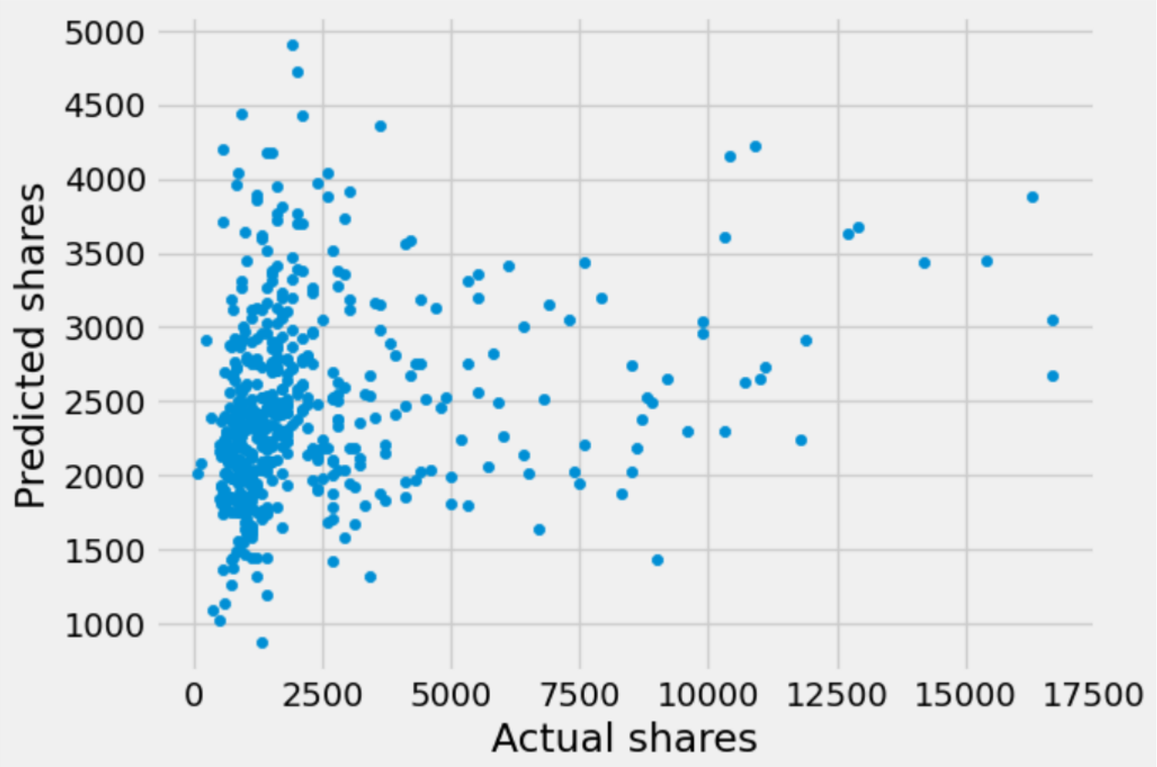
We have a 80% Train set with 29 976 rows and a 20% Test set with 7494 rows.

Modeling

#1. Linear Regression

We test our LinearRegression model with a sample of 500 values from the training set.

	Actual shares	Predicted shares
0	3100	1670.995200
1	1500	2860.915933
2	1900	2715.613544
3	1500	2547.820292
4	773	2923.400237

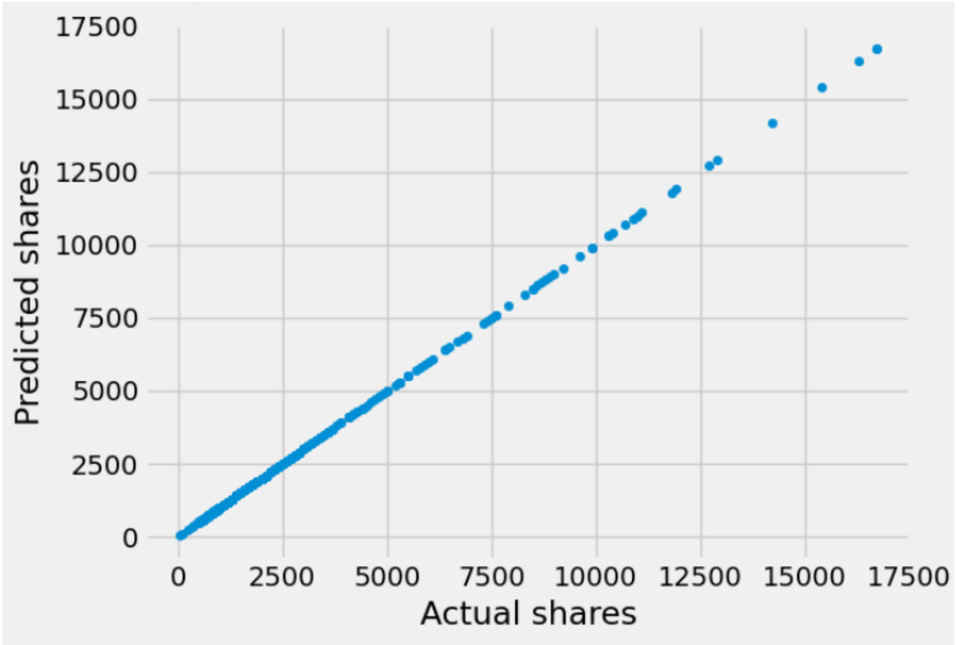


Modeling

#2. Decision Tree

We test our LinearRegression model with a sample of 500 values from the training set.

	Actual shares	Predicted shares
0	3100	3100.0
1	1500	1500.0
2	1900	1900.0
3	1500	1500.0
4	773	773.0



It seems that the predictions have zero error but we gonna **compute the RMSE and the MAE to confirm.**

Verify
Decision Tree model

RMSE & MAE

```
RMSE : 0.0
MAE :  0.0
Median : 1400.0
```

The results seem to indicate that we have a perfect model. It's called **overfitting** and it can be bad because it can provide a bad prediction on the test dataset because he will have never seen it before.

We are going to do a cross-validation to test the performances of the model.

Cross-validation

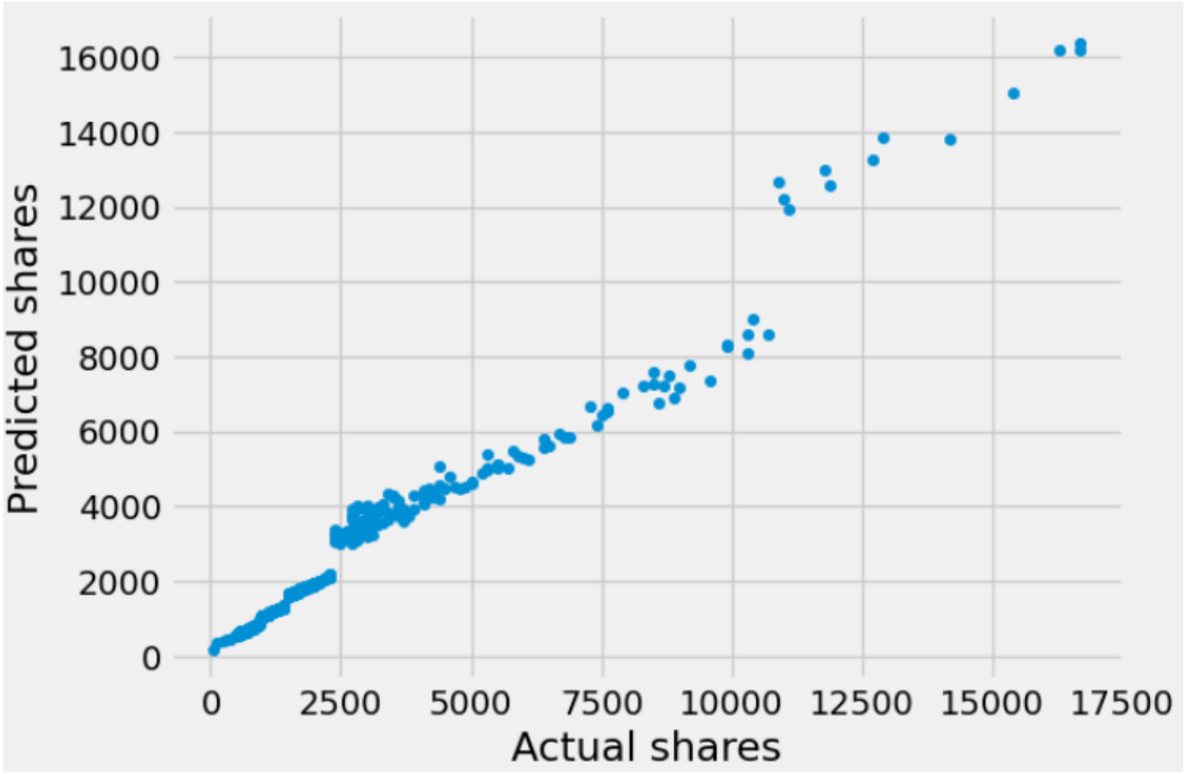
```
SCORES : [1647.10601348 1608.36477451 1656.59812385 1631.23410309
1584.6107889 ]
MEAN :  1625.5827607657893
STANDARD DEVIATION :  26.200390663002427
```

So we can see that the model is not as good as expected.

Modeling

#3. Random Forest

	Actual shares	Predicted shares
0	3100	3253.00
1	1500	1639.00
2	1900	1873.00
3	1500	1611.00
4	773	755.33



Verify

Decision Random Forest

RMSE & MAE

```
RMSE : 433.90811012758326
MAE :  224.7200490392314
Median : 1400.0
```

Just in case, we are goig to do a cross-validation to be sure it's not a overfitting case :

Cross-validation

```
SCORES : [1168.30927982 1139.70444033 1149.47417059 1173.79331164 1152.93735353]
MEAN :  1156.8437111822661
STANDARD DEVIATION :  12.506578938704592
```

After cross-validation, the results are getting worse.

Summarizing the 3 models

Pre-cross validation

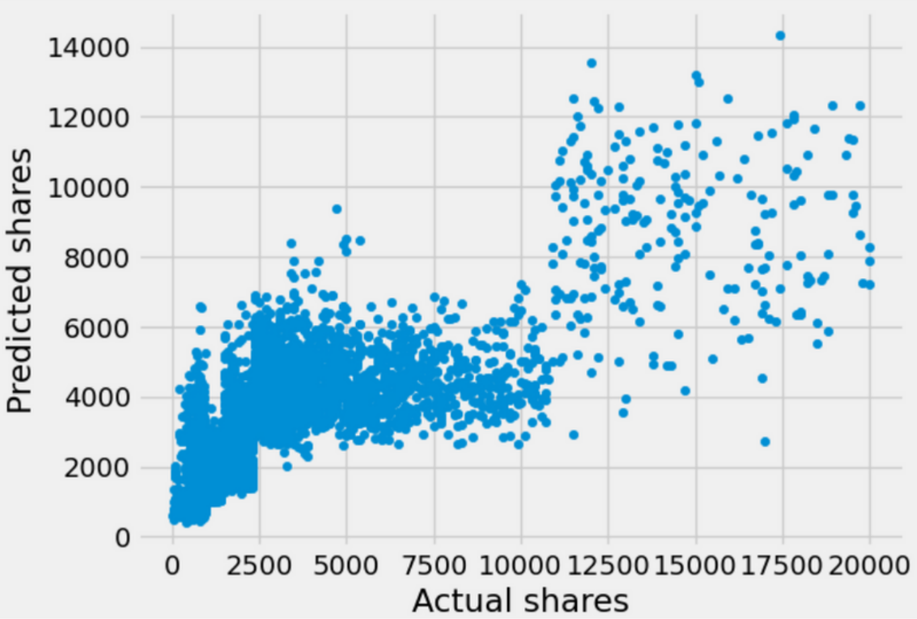
```
Linear Regression :
    RMSE =  2 728.27
    MAE = 1 709.03

Decision Tree :
    RMSE = 0
    MAE = 0

RandomForest :
    RMSE = 433.91
    MAE = 224.72
```

Overfitting

The best model !!!

Evaluating the final model
with the test set

We test the **Random Forest** model on the other part of the data set



	Actual shares	Predicted shares
0	6700	4690.0
1	4100	3394.5
2	6200	2771.5
3	1700	1930.0
4	721	4034.3

Creation of a dataframe to show results

We can see a pattern but the results are not particulary good. There is a lot of dispersion.

API

Our API takes **3 inputs** : a title, a text, and a categorie
It then uses our model and gives you a prediction whether It will be popular or not.

Conclusion

Correlations between our variables and our target is low --> tough to find relevant key variables to study

Random forests regression is the best estimator for our problem, but we have debatable results

Upgrades in computational power could allow us to run tests with more parameters, which might improve our results

More ML models ?

Fixing more outliers