Project - Group 109

LELEC2870 - Machine Learning: Regression, Deep Networks and Dimensionality Reduction

Gabin Kana Tsiguia (05581900, DATSM)
&
Hervé Azobou Anantia (02411200, STAT2FC)

December 19, 2020

1 Introduction

News has become a part of our daily life. Tons of publishers release thousand online articles every single day. Before the publication, it is quite difficult to know how many times an article will be read and shared. Therefore, predict a popularity of an article before its publication could be valuable both for authors and publishers.

In this project, we will use the data collected by *Kelwin Fernandes*, *Pedro Vinagre* and *Paulo Cortez* to predict the popularity (number of shares) of online articles based on their components.

We will use the Pyhton programming language and the $scikit - learn$ package API for all our model implementation. For this work, the estimate the score we expect is 0.50.

2 Data description

2.1 Target variable

Our target variable represents the number of times each Mashable article in the dataset has been shared during the period of data collection. That number of article's shares varies from 1 to 843300 in the training dataset. The average number of shares is around 3430. Among all articles in the training data, 25% have been shared at most 949 times while 50% have been shared at least 1400 times. Only 5% of the articles have been shared at least 11 000 times. Less than 1% of the articles have been shared at most 45 times and less than 1% at least 19 7600. The target variable seems to vary a lot, the sample standard deviation is more than three times greater than the sample mean. The distribution of the target is very right skewed. This is certainly due to the fact that 75% of the articles have been shared at most 2 800 times while few have been shared up to 300 times the previous number.

2.2 Features

- There are features with a minimum value of zero meaning some articles don't have certain characteristics (number of words in the article, average word length, the number of unique tokens, etc.)
- Some features have a quite symmetric distribution (number of words in the title($n\_tokens_{t}itle$). It also would be the case for some variables if the value zero were not present. Those variables have a pic at 0 but the rest of the distribution seems to be symmetric (Average word length (*average_token_length*), rate of unique words (*n_unique_tokens*), article subjectivity (*global_subjectivity*), etc.).
- On the other hand, many features have heavy right skewed distribution with the mode closer to zero. Some examples are: number of words in the article (*n_tokens_content*)), number of links (*num_hrefs*), number of images (*num_imgs*), etc.
- The rate of non-stop words (*n_non_stop_words*) is supposed to range from 0 to 1 but most of the values are greater than 0.99).
- Most of the articles reference Mashable articles that have no shares
- Most of the articles are published on Tuesday, Wednesday and Thursday. In consequence, fewer are published on weekends compared to working days.
- One of the most important remark is that the features all have different scales and some articles do not have any content (no article text, no images and no videos).
- There are no missing values the whole data set (features and target).

2.3 Relationships

- The target variable is not linearly correlated with any features. All the Pearson correlations (in absolute value) are less than 0.10. This does not mean that there is no relation between the target and the features, it just means that relation is not linear. This remark is pretty much the same regarding the rank correlations (Kendall and Spearman) where the highest value is 0.26 for Spearman rank correlation and 0.17 for Kendall.

- However some numeric features are highly and positively correlated with each other. It is specially the case for the **Word related features**, the **Links related features** and some of the the **Keywords related features**. For example the linear correlation between the average word length (*average_token_length*) and the rate of non-stop words (*n_non_stop_words*) is 0.94. Some of the **Natural Language Processing features** are moderately correlated with some of the **Word related features** (Figure 2)

3 Modelling

In order to best evaluate our final model, we split the whole dataset into two sets: 80% for the training set and 20% for the test set. In this work, we have implemented the following regression models:

1. Ordinary Least Squares (OLS): classic multiple linear regression
2. Stochastic Gradient Descent Regression (SGD)
3. K-Nearest Neighbours (KNN) Regression
4. Lasso Regression
5. Multi-Layer Perceptron (MLP) regression
6. Random Forest (RF) regression

3.1 Pre-processing

It is well known that most of the machine learning algorithms require that the features have some specific characteristic in order to work properly. During the exploratory data analysis, we notice that features all different scales and different variability. For these reasons, we applied the following preprocessing steps 1. The log transformation has been applied to positive unbounded numerical features to help handle their skewed distributions by making it approximately bell shaped and by reducing the order of magnitude. That transformation also decreases the effect of outliers. 2. All the numerical features have been standardized to become zero-mean and unit-variance

3.1.1 Model comparison and first predictions

To compare all the models we have, we used the provided score function which is the average of four $F1 - scores$. We trained all the models using both raw and preprocessed data. It appears that the Random Forest seems to have the best prediction score while the others are quite similar in terms of prediction score. However, the SGD regression model is the one with the lowest prediction score. The preprocessing seems to have a small positive impact on the capability of each model.

	OLS	SGDR	Lasso	KNN	MLP	RF
On raw data	0.489	0.494	0.490	0.588	0.441	0.833
On processed data	0.490	0.302	0.488	0.597	0.487	0.832

3.2 Model Selection

In order to select the best promising model, a cross validation was performed. The cross validation allows the estimation of the prediction capability of the model on unseen data. We used a $5-fold$ cross validation which consist of dividing the training data into 5 parts, then training each model 5 times using 4 folds and validating using the remaining fold. The table below shows that the Random Forest model has the highest score on both training and validation sets, followed by KNN and the MPL models. The SGD regression still has the lowest score. However there is a huge difference between the score obtained on training set and the one obtained on validation set for the Random Forest model. This could be a problem of overfitting that needs to be investigated.

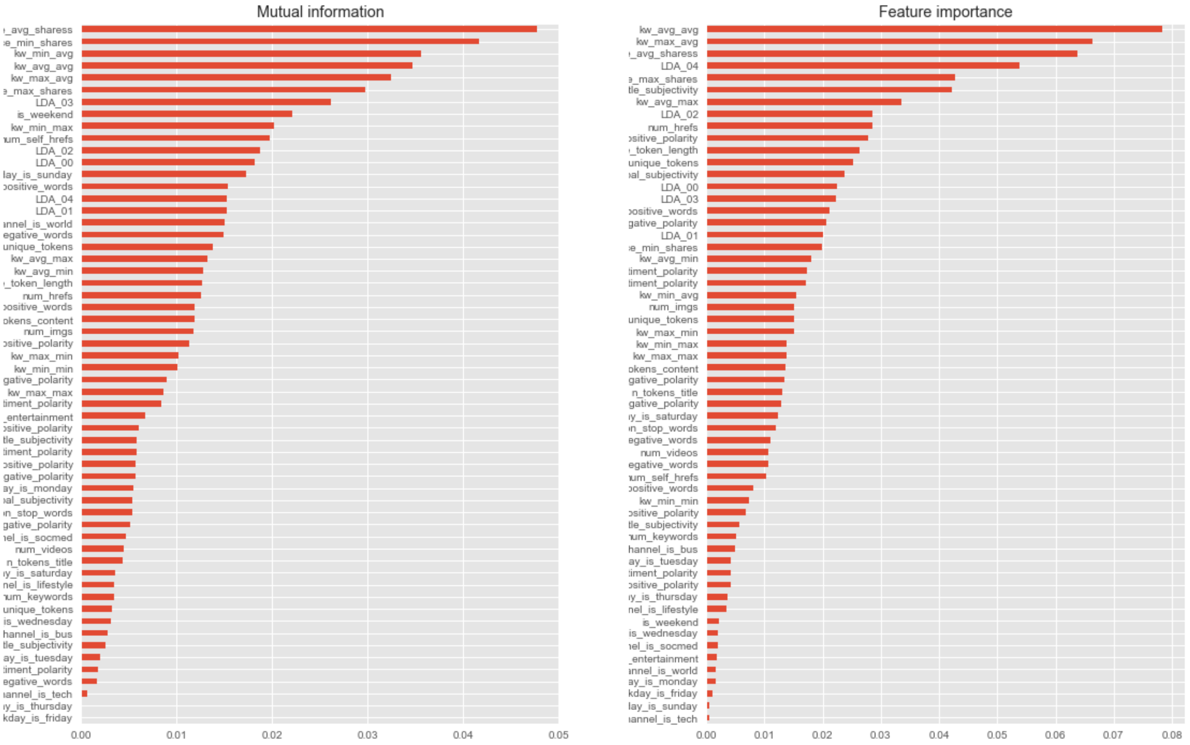
	OLS	SGDR	Lasso	KNN	MLP	RF
Training	0.4902	0.2384	0.4898	0.5886	0.4846	0.8280
Validation	0.4838	0.2410	0.4846	0.4704	0.4776	0.5246

After cross validation, the models we retain and will further improve to get the final predictions are KNN and Random Forest. Random Forest has the highest scores on both training and validation sets. KNN has higher on training set and similar score as the MLP on validation sets.

3.3 Feature selection

To select the most relevant features in order to improve the model we choose, we use the mutual information between the target (number of shares) and the features. We used that method because during the EDA, we noticed that there was no strong linear correlation between the target and the features. As a second way to confirm the previous, we also used used feature importances provided by a Random Forest model with 200 Decision Trees. When observing the graphs below (Figure 1), one can notice the top $30-45$ features are almost the same for both methods. For that reason we selected the common variables among the top 45 features selected by each method. Most of the selected features are keyword features (*number*). We also have many Natural language processing features as well as few words features.

Figure 1: Mutual information and feature importances



3.4 Model optimization and final estimator

To try optimizing both KNN and RF models, we adopted a grid search. For KNN we focused our attention the -number of neighbours-, the -*metric* (*distance*)- hyperparameters. For Random Forest, we considered the -*maximum depth*- of each Tree, the -number of Decision Trees- hyperparameters. We first ran experiments to get the best value of each hyperparameter. Those experiments consisted of plotting the scores against each hyperparameter (See the validation curves in appendices figures 6-9). After the experiments we adopted the following grid: For Random Forest ($maximum\,depth \in \{20, 23, 25, 30\}$ and $number\,of\,Trees \in \{150, 200, 300\}$) and KNN ($number\,of\,neighbours \in \{3, 5, 7, 9, 11, 13\}$ and $metric \in \{minkowski\,, mahalanobis\,, seuclidean\}$). After the grid searches, the best model (model with highest score on validation set) was the Random Forest with a maximum depth of 23 and 150 Decision Trees.

4 Conclusion

The final model we used to get predictions is Random Forest Regression. With that model, we obtained a score of 0.5044 on test data we saved. Thus we estimate that we will get a score of 0.50 on the new data.

This project has been a way for us to learn and practice machine learning algorithms. We did not go deep into some aspects that could allow us to get better score because time consuming or some require more computational power we did not have. Those aspects include feature engineering, model selection and optimization. It would have been worth to investigate the MLP model and Lasso.

Appendices

Figure 2: Top Pearson Correlations

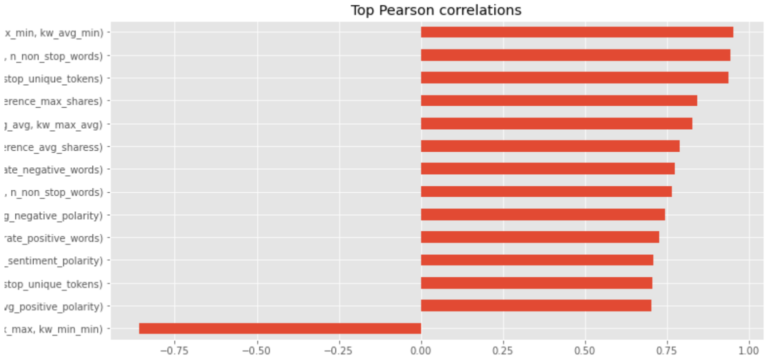


Figure 3: Boxplots of Log-Shares by article category

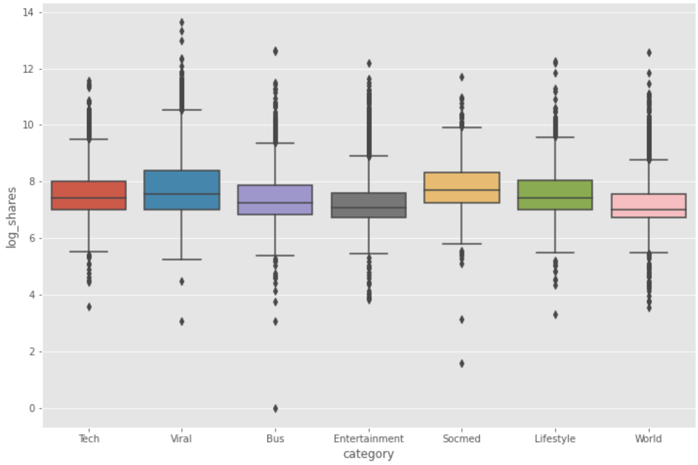


Figure 4: Boxplots of Log-Shares working vs weekend

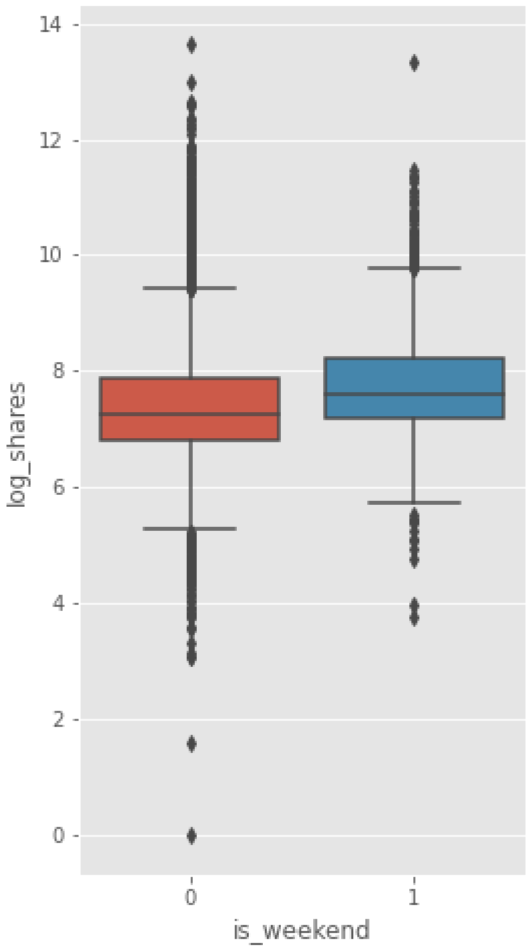


Figure 5: Training and validation scores

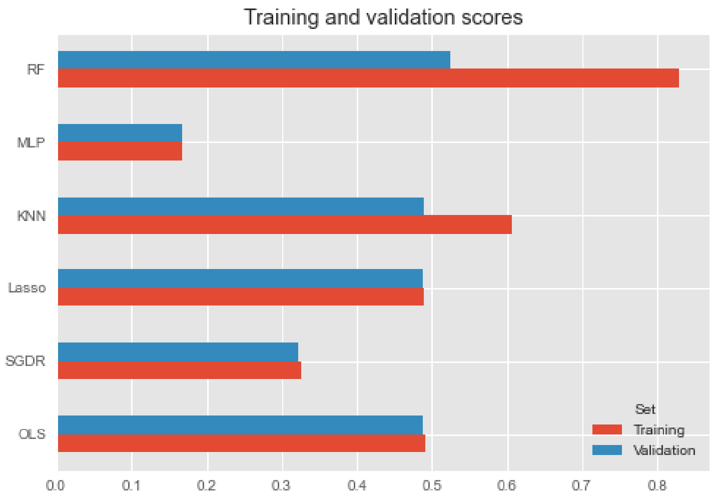


Figure 6: Validation curve number of neighbours KNN

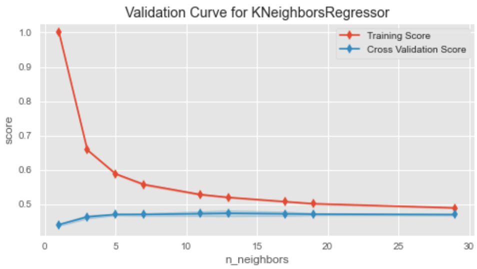


Figure 7: Validation curve Weights KNN

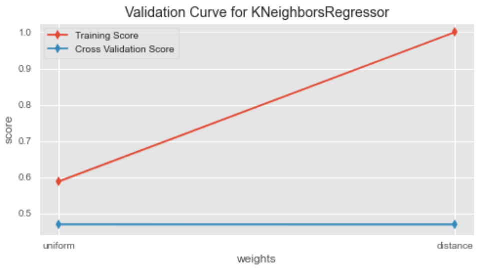


Figure 8: Validation curve of the maximum depth - RF

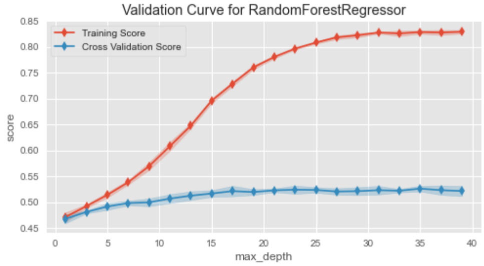
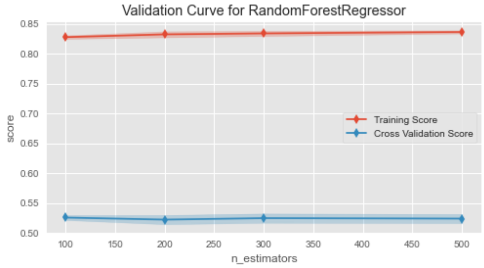


Figure 9: Validation curve of the number of Decision Trees - RF


Validation Curve for RandomForestRegressor
Training Score
Cross Validation Score
score
n_estimators