LELEC2870 Project
Predicting Shares of Articles

Group 90
Cédric Antoine, Grégoire Vekemans

Hand in: December 2, 2021

Introduction

This project was carried out as part of the LELEC2870 course (Machine Learning: regression, deep networks and dimensionality reduction) at UCLouvain. The objective here is to develop several regression models which predict the number of shares of an online article.

Pay attention, at the very beginning of the project we split the dataset into a training/validation set and a test set (that we will use later to evaluate the final model) The results we obtain may slightly vary due to this split (done randomly).

1 Data cleaning

After observing some initial regression results, we wondered if removing outliers and overly redundant features might not help us increase our regression scores. We then implemented a method that removed all lines for which $X_i < \overline{X} - 4\sigma^2$ or $X_i > \overline{X} + 4\sigma^2$; it then leads us to remove 4000 lines approximately.
Furthemore, in respect to avoid any overfit in the regression model one decides to remove some features of his datas. More some features correlation are close to each other, more variables inside those features are too. We were then curious about the impact of the proximity and decide to keep just one and remove others which are too close ($\text{cov}(X_i, X_j) > 0.999$).

However, after performing these operations we observed a slight decrease in our scores. By observing the plots of our data we realised that they varied greatly and that there were therefore a lot of outliers (Fig. 6). We believe that this decrease in scores is precisely due to this. Indeed, if we train the models without these outliers, the model will then be unable to correctly manage outliers in the data that we will provide for prediction.

This is why we have decided not to use downsizing and not to remove outliers for the final model training process.

2 Features Selection

One decides here to test three different approches. In this section we briefly present 3 methods. By doing so we are then able to compare different feature selection and see impact of them on the regression models, whether they are good or bad. We will then be able to discuss about the model selection in Section 3.

2.1 Correlation

Correlation is obviously an easy way to select the features. However we have seen during lectures that it was not a very efficient method because of its linear approach. We decide in that respect to only use correlation as feature reduction for the linear regression as it still seemed appropriated for this model.

2.2 Principal Components Analysis

The second chosen approach is the Principal Components Analysis (PCA). It shows out that for keeping a minimum of 0.95 of the inital variance we would be forced to keep 36 features (see Fig. 1). But using such a lot of features may cause the computations very slow. This is the reason why we decide to focus ourselves on a more limited amount of features as it is explained in Section 3.

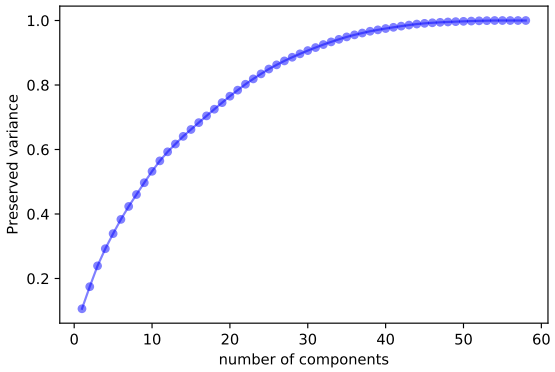


Figure 1: Preserved Variance of PCA

2.3 Mutual Information

The last features selection method we choosed to use is the mutual information. This method gives good results starting from around 20 features.

3 Models selection

For the model selection we implemented a method based on brute force and K-fold cross-validation. Indeed our algorithm runs through a set of hyperparameters (for loops) and realizes for each combination of hyperparameters a regression for which the score is obtained by the

average of the scores obtained in the cross validation. By doing so we increase the probability to have scores that are not only due to chance after a random choice about train and validation sets.

Please note that we willingly restrain the amount of studied parameters in aim to reduce the computational time ; indeed, an exponentially increasing time complexity would result from increasing the model parameters (but could obtain better results).

3.1 Linear Regression

The linear regression model is naturally the simplest one. The only parameter we could play with is the amount of features : how many features will we keep in aim to optimize the score ? One has represented on Fig. 2 the regression model for the three features selections discussed in Section 2. We directly observe that PCA lacks for accuracy. Mutual information has quite good results but needs a lot of time to be performed in front of the covariance features selection which has nearly the same results. In addition to that, correlation is a linear operation such that we obtain results we were expecting. We observe in that respect a maximal score with an amount of 10 features approximately.

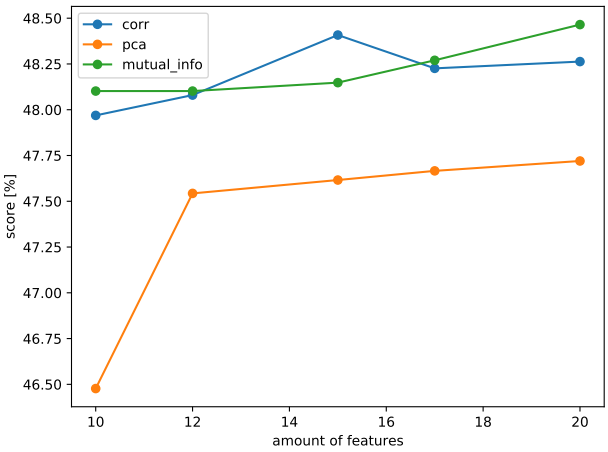
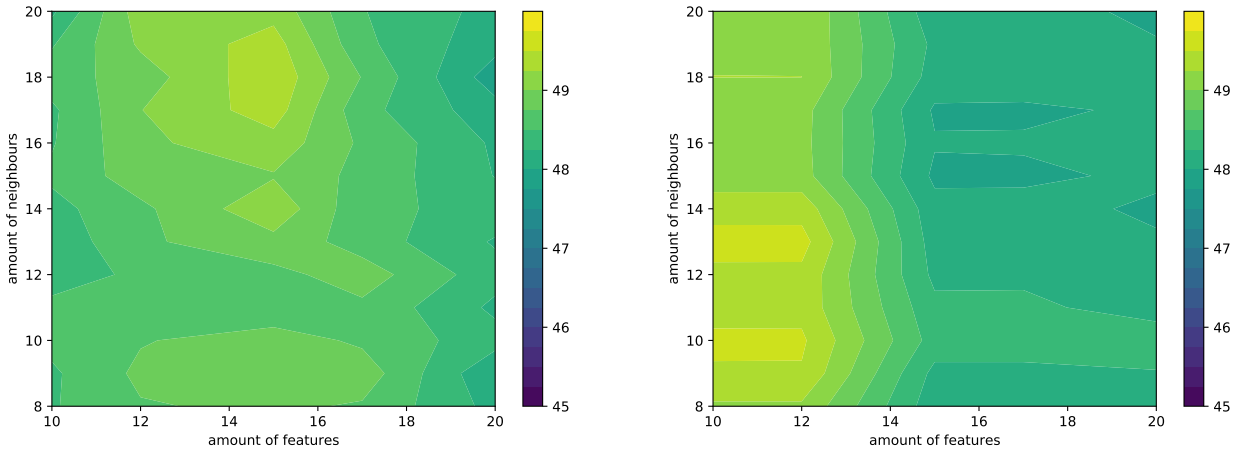


Figure 2: Score of the Linear Regression Model for Different Features Selection

3.2 k-NN Regression

Next we look to the k-NN regression model. Hyperparameters are the amount of features and the amount of neighbours. One plotted the score variation for both the PCA and the mutual information features selection on Fig. 3. We first remark on the two plots that score is highly dependant to the amount of features : we conclude to an optimal $n_{\text{features}} \approx 15$ for Fig. 3a and $n_{\text{features}} \approx 10$ for Fig. 3b. On Fig. 3a, the score variation according to $n_{\text{neighbours}}$ is more random but we easily observe a maximum at $n_{\text{neighbours}} = 18$. Concerning Fig. 3b it seems rather constant for high amount of features. Yet for $n_{\text{features}} = 10$ we conclude to an optimal $n_{\text{neighbours}} = 10$ or 12. We conclude for this section that both models are good in front of the linear models studied earlier and equivalent to each other. Even if PCA features selection

offers a higher values, we know that regression model to be more performant around 15 to 20 neighbours.



(a) PCA (b) Mutual Information

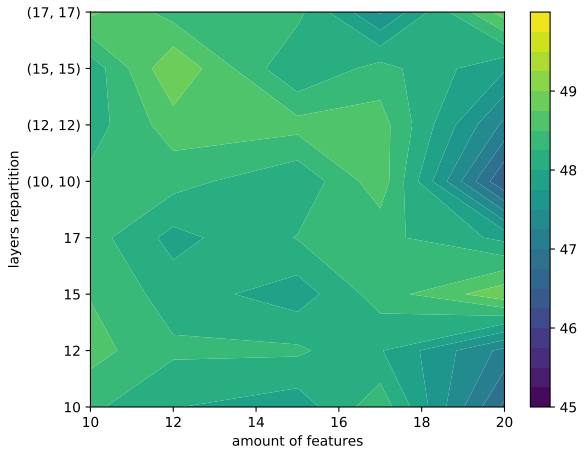
Figure 3: Score [%] of the KNN Regression Model for Different Features Selections

3.3 MLP Regression

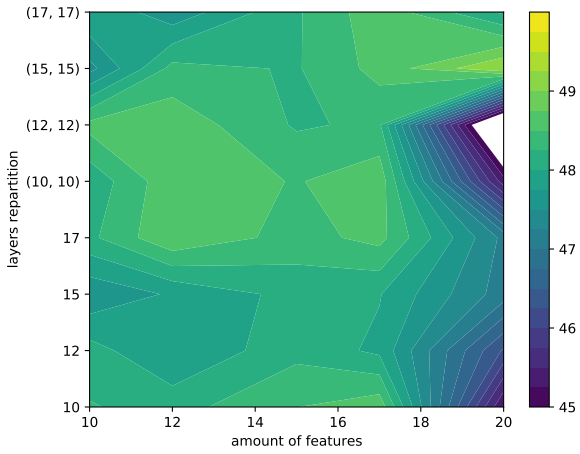
The last model we were given is the MLP regression models. There are here much more hyperparameters such that we choose to plot just the best ones. In that respect we observed that mutual information feature selection preprocesses much better the data compared to the covariance and the PCA methods. We then studied the impact of the amount of features, the layer repartition such as the learning rate. The associated plots are shown on Fig. 4. If we observe these figures we can see that there is no real logical distribution of the score. We note that for the inverse scalling learning rate and between 18-20 features, we obtain really low scores. We also see that for the adaptive learning rate the score is pretty much stable between 14-18 features. But on a bigger scale it is difficult to draw any conclusion from these graphes. We only observe that compared to the KNN results the MLP results are worse and more unpredictable.

3.4 PLS regression

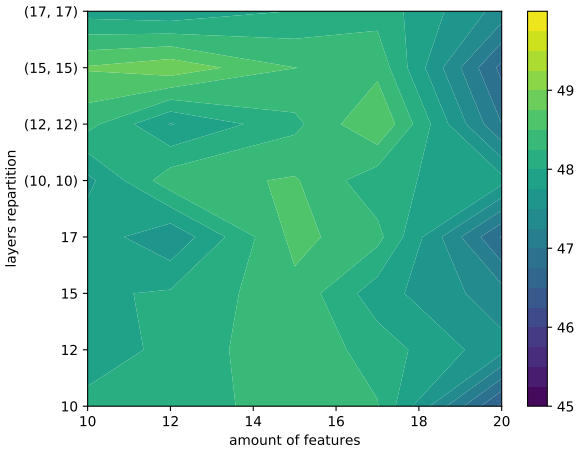
Finally, we choose as the last regression model the Partial Least Square (PLS) algorithm. While the linear regression finds hyperplanes of maximum variance between the response and independent variables, the PLS method finds a linear regression model by projecting the predicted and the observable variables to a new space. The only parameters we could play with in addition to the amount of features is the amount of components. We represent the score variation with respect to those two parameters on Fig. 5. For the PCA features selection, we observe that the score stays constant whether the amount of components we modelize it. Moreover, the score also stays constant starting from 12 selectionned features, and score values for $n_{\text{features}} = 10$ is smaller. Switching to Fig. 5b we observe an increasing score with both an increasing amount of features and components. Even if the scores are better as for a PCA features selection, we generally observe for PLS regression models bad results in front of previous sections.



(a) Constant

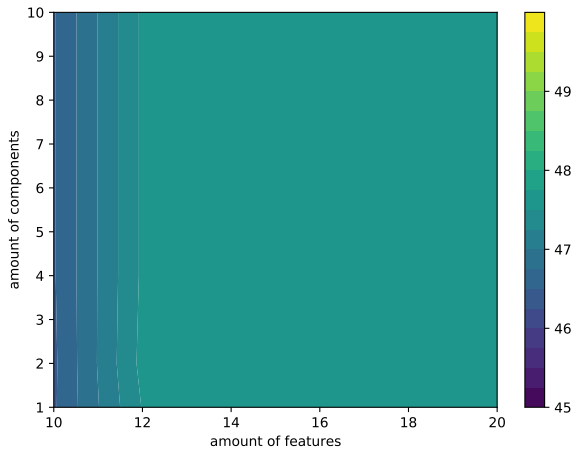


(b) Inverse Scaling

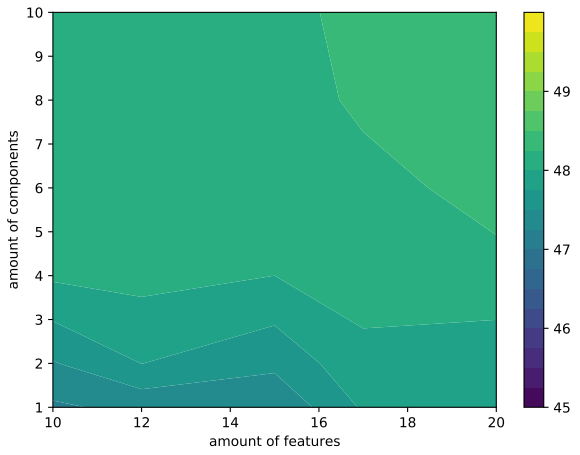


(c) Adaptive

Figure 4: Score [%] of the KNN Regression Model after a Mutual Information Features Selection for Different Learning Rates



(a) PCA



(b) Mutual Information

Figure 5: Score [%] of the PLS Regression Model for Different Features Selections

4 Final Model

In order to choose the final model we started by observing all the scores obtained by each model in the model selection. We saw that two models stood out.

First, the KNN regression model with 18 neighbours and the PCA as feature selection with 15 features. With this model we obtained a score of 0.49444. Secondly, the KNN regression model with 13 neighbours and mutual information as feature selection with 10 features. With this model we obtained a score of 0.49599.

To choose between these two models we took two factors into account. Firstly the number of neighbours used. Previously we have observed that overall KNN regression works best around 15-20 neighbours. This led us directly to the first model. Second, we took into account the type of feature selection. By comparing PCA and mutual information, it immediately appeared to us that PCA took much less computation time.

So we chose the KNN regression with 18 neighbours and with PCA as feature selection and 15 features as final model despite a lower score than the second one.

In order to evaluate this model we finally used the test set that we had set aside at the start of the project.We trained the model with the entire validation/train set that we before used to train and validate our models. Then made some predictions with the test set to estimate his performaces.

5 Conclusion

We can conclude by saying that KNN is in our case by far working best. Even if we can sometimes obtain interesting results with the MLP regression these seem not really stable. Finally we estimate that our final model will have a performance of between 0.485-0.495 for the obtained score.

6 Appendix

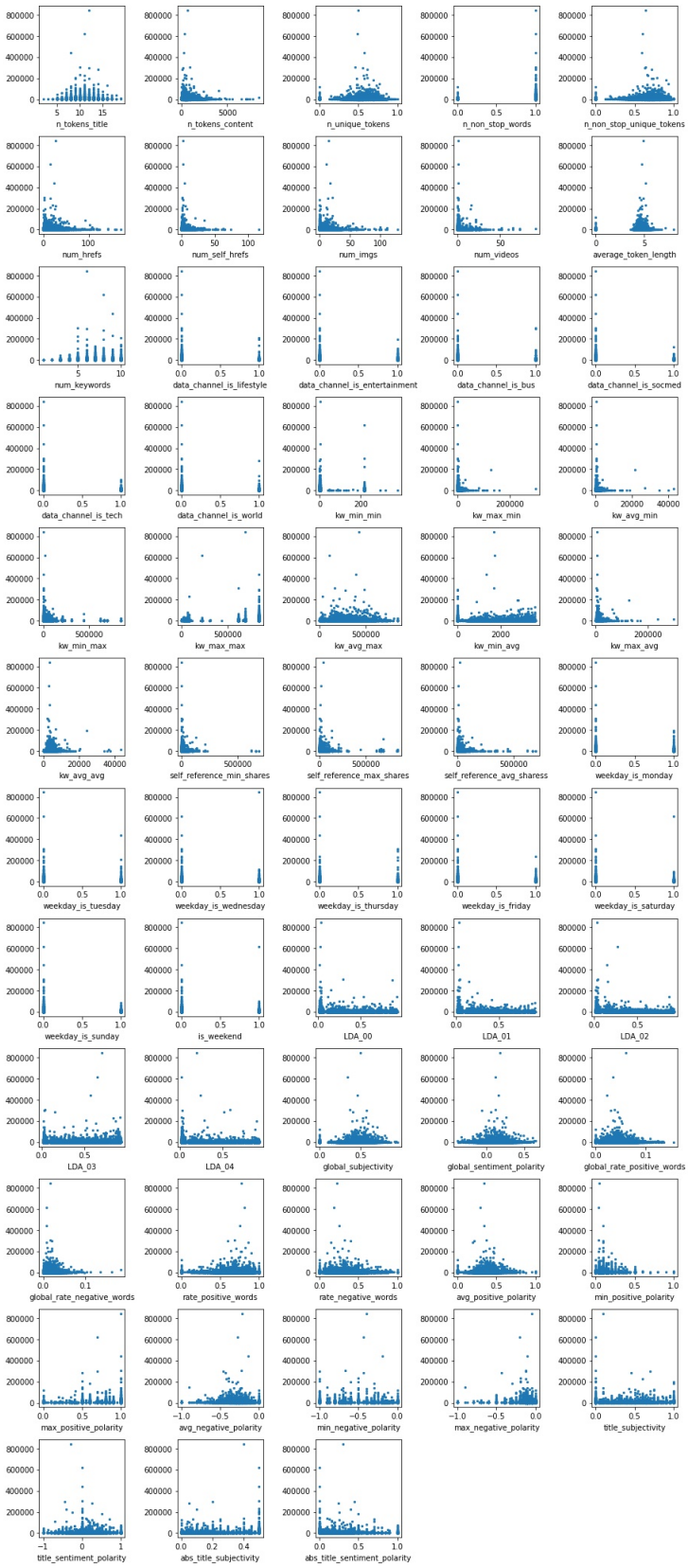


Figure 6: Representation of Datas for all Features