ML Assignment Report

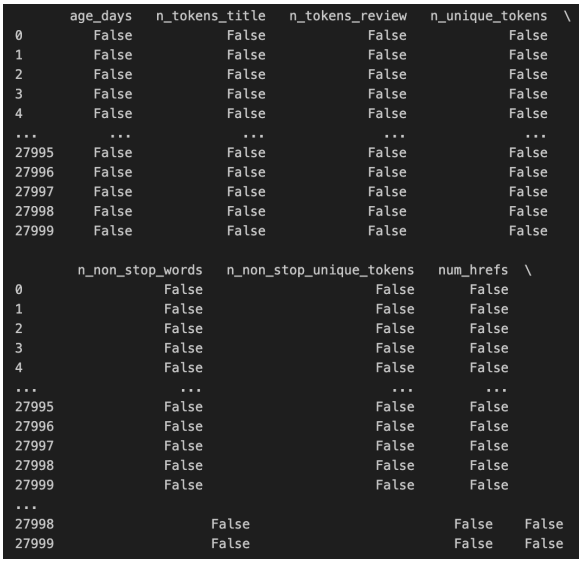
Juan Ocampo, Juan Saavedra, Ziad El Harairi

In this project, we are developing a regression model to predict customer satisfaction through popular reviews which is defined by reviews collecting a large number of shares. Through this model we will try to study the characteristics that make a review to be popular, which could help in an early intervention in case of popular negative feedback.

Our data includes textual features of the reviews, for example the number of words, the subject , and the sentiment associated with the title and content. We will therefore use these textual features to predict the amount of shares reviews will have in the future.

Pre-processing

In our preprocessing phase we started by looking at our data and checking for null values and how big is our data. From preprocessing, we found that we have 28000 rows and 39 columns in our previous data which we will use to train and test the model. We then checked through the data to find null values in any of our features. As you can see below we do not have any null values in our data.



	age_days	n_tokens_title	n_tokens_review	n_unique_tokens	\
0	False	False	False	False	
1	False	False	False	False	
2	False	False	False	False	
3	False	False	False	False	
4	False	False	False	False	
...	...	...	...	...	
27995	False	False	False	False	
27996	False	False	False	False	
27997	False	False	False	False	
27998	False	False	False	False	
27999	False	False	False	False	

	n_non_stop_words	n_non_stop_unique_tokens	num_hrefs	\
0	False	False	False	
1	False	False	False	
2	False	False	False	
3	False	False	False	
4	False	False	False	
...	...	...	...	
27995	False	False	False	
27996	False	False	False	
27997	False	False	False	
27998	False	False	False	
27999	False	False	False	
...				
27998	False	False	False	
27999	False	False	False	

In our data we have many features, we have 38 columns that have numerical and categorical features. The numerical columns measure different aspects about our reviews, therefore they will not have the same scale. As a result, we had to standardize our numerical features. This

means that we transform our values to have a mean of 0 and standard deviation of 1 as to not give more weightage to specific features with high values. This step is done by calculating the mean and standard deviation of each feature and then subtract each data point by the mean and divide by the standard deviation. For this we use the python library StandardScalar.

$$z = (x - \mu)/\sigma$$

Our data includes categorical features, some of these are “day” and “product category”. We have “day” as ordinal categorical data because they could be ordered as days of the week. On the other hand, we nominal categorical data because they can not be ordered. For creating our model we will have to separate the categorical values from the numerical in a separate dataframe.

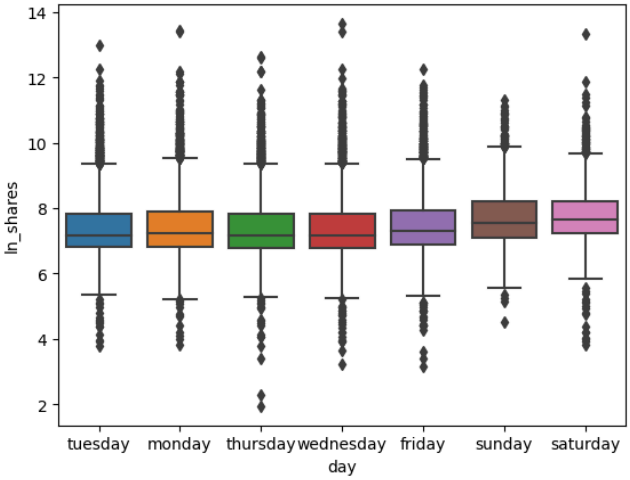
- Multicollinearity in categorical data (if we have it)

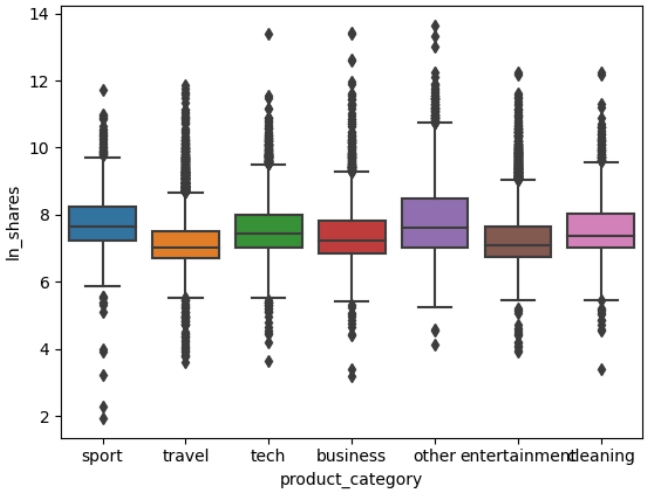
Visualization

For our visualization and exploratory data analysis we started by transforming our data into numbers and looking at the statistical summary of the features. As seen below, each feature has a different mean, standard deviation and interquartile range because we did not standardize the data yet.

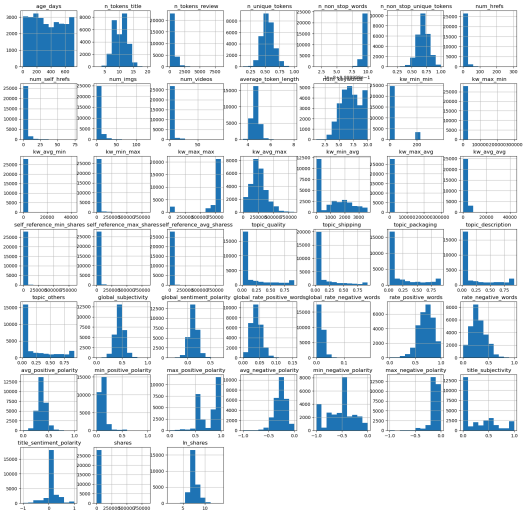
	count	mean	std	min	25%	50%	75%	max
age_days	28000.0	360.376286	2.125589e+02	8.000000	175.000000	348.000000	547.000000	731.000000
n_tokens_title	28000.0	10.379679	2.106580e+00	2.000000	9.000000	10.000000	12.000000	20.000000
n_tokens_review	28000.0	559.612036	4.640245e+02	18.000000	258.000000	420.000000	728.000000	8474.000000
n_unique_tokens	28000.0	0.547309	1.024567e-01	0.123422	0.477454	0.543549	0.611805	1.000000
n_non_stop_words	28000.0	1.000000	4.052939e-09	1.000000	1.000000	1.000000	1.000000	1.000000
n_non_stop_unique_tokens	28000.0	0.693451	1.010963e-01	0.119134	0.632627	0.693603	0.757404	1.000000
num_hrefs	28000.0	11.229857	1.142456e+01	0.000000	5.000000	8.000000	14.000000	304.000000
num_self_hrefs	28000.0	3.382214	3.812180e+00	0.000000	1.000000	3.000000	4.000000	74.000000
num_imgs	28000.0	4.558214	8.275950e+00	0.000000	1.000000	1.000000	4.000000	128.000000
num_videos	28000.0	1.261143	4.214034e+00	0.000000	0.000000	0.000000	1.000000	91.000000
average_token_length	28000.0	4.689296	2.838252e-01	3.600000	4.497788	4.675410	4.863636	8.041534
num_keywords	28000.0	7.211250	1.914299e+00	1.000000	6.000000	7.000000	9.000000	10.000000
kw_min_min	28000.0	26.591536	7.016289e+01	-1.000000	-1.000000	-1.000000	4.000000	377.000000
kw_max_min	28000.0	1128.479376	3.452898e+03	0.000000	445.000000	662.000000	1000.000000	298400.000000
kw_avg_min	28000.0	309.609435	5.416904e+02	-1.000000	142.833333	238.000000	360.166667	42827.857143
kw_min_max	28000.0	13094.884464	5.678621e+04	0.000000	0.000000	1400.000000	7700.000000	843300.000000
kw_max_max	28000.0	750165.917857	2.169170e+05	0.000000	843300.000000	843300.000000	843300.000000	843300.000000
kw_avg_max	28000.0	255463.140848	1.319872e+05	0.000000	171490.500000	242474.857142	327190.000000	843300.000000
kw_min_avg	28000.0	1104.204114	1.126548e+03	-1.000000	0.000000	1014.509912	2028.768897	3610.124972
kw_max_avg	28000.0	5583.015923	5.996637e+03	0.000000	3551.066558	4307.228118	5950.000000	298400.000000
kw_avg_avg	28000.0	3102.824657	1.295372e+03	0.000000	2376.593730	2852.712114	3545.119455	43567.659946
self_reference_min_shares	28000.0	4156.961040	1.978049e+04	0.000000	705.000000	1300.000000	2700.000000	843300.000000
self_reference_max_shares	28000.0	10583.641114	4.180771e+04	0.000000	1200.000000	3000.000000	8200.000000	843300.000000
self_reference_avg_sharess	28000.0	6589.157040	2.439879e+04	0.000000	1100.000000	2300.000000	5300.000000	843300.000000
topic_quality	28000.0	0.188569	2.658784e-01	0.018182	0.025064	0.033433	0.254978	0.920000
topic_shipping	28000.0	0.142482	2.210914e-01	0.018182	0.025015	0.033347	0.151288	0.925947
topic_packaging	28000.0	0.215686	2.814069e-01	0.018182	0.028571	0.040005	0.330399	0.919999
topic_description	28000.0	0.215093	2.895497e-01	0.018182	0.025661	0.040000	0.342517	0.926534
topic_others	28000.0	0.238170	2.916853e-01	0.018182	0.028575	0.050000	0.413203	0.927076
global_subjectivity	28000.0	0.457169	8.833106e-02	0.000000	0.402383	0.456943	0.510549	1.000000

As a first step we started by creating box plots to see the distribution in our categorical variables against our target variable.





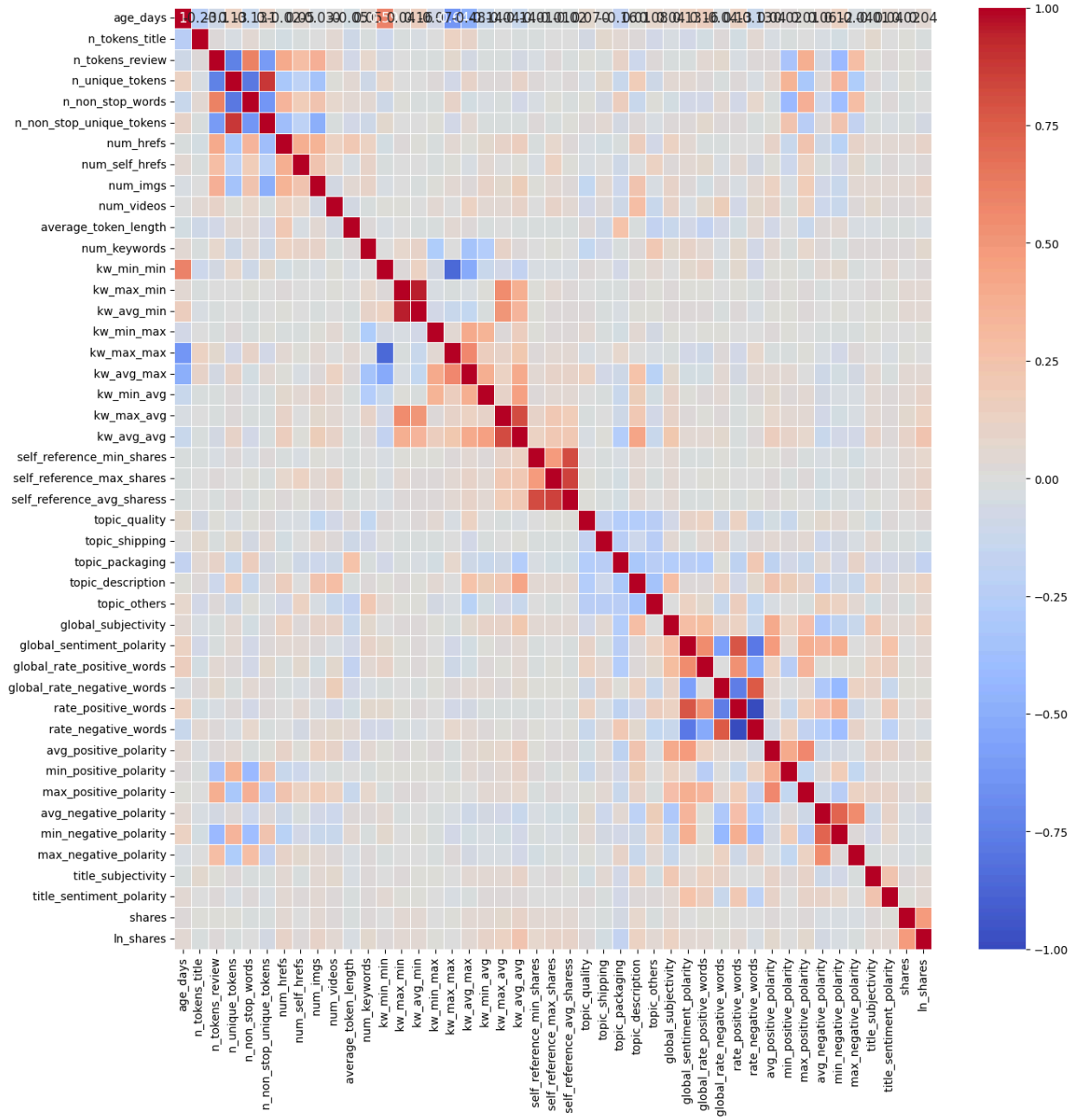
Our first boxplot shows the distribution of each day to the logarithm of shares. We are using the logarithm of the shares and not the shares to learn by what amount relative to a base share value will there be a change based on the day of the week. We implemented the log transformation of the target variables to make it as “normal” as possible.



We then used histograms to plot the distribution of numeric features. We can see that we have various types of distributions, we first see that many distributions are right skewed. Others have a normal distribution with a slight skewness. We also see features, such as “min_negative_polarity” and “kw_min_avg”, with distributions that are not normal.

Correlations

We checked the correlations to find interactions between variables that we may find useful to explain the number of shares and log of shares. In this case, it is possible to observe some correlations that get accentuated due to the log transformation, where the untransformed feature shows a low correlation with all features.

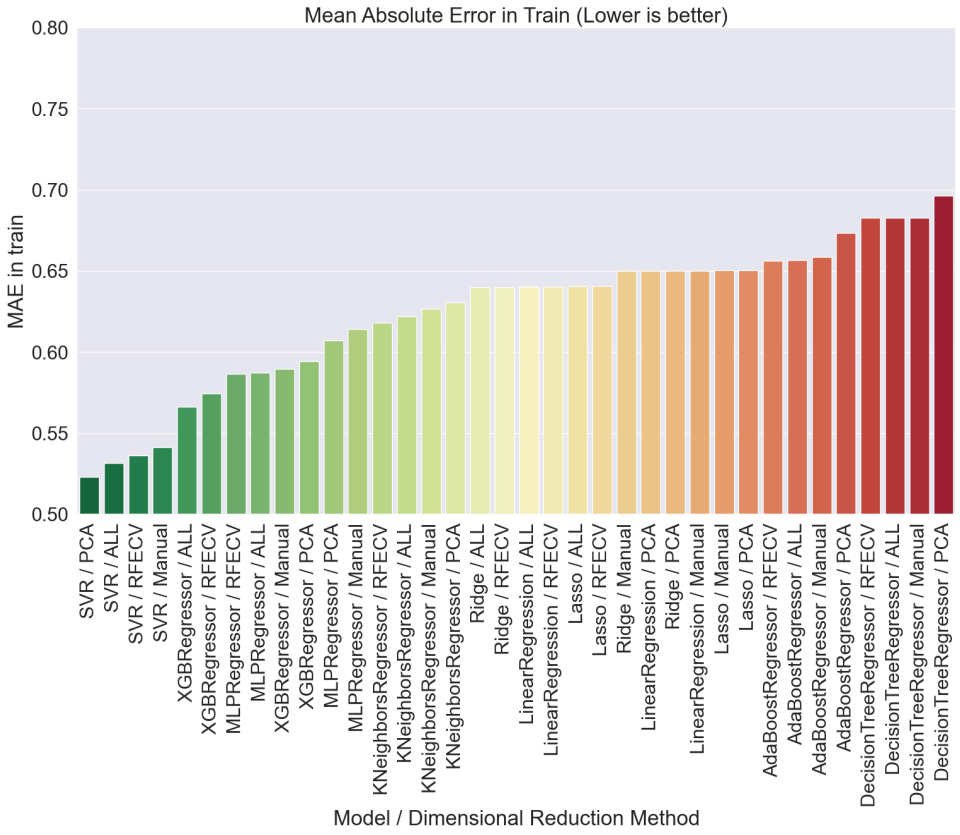


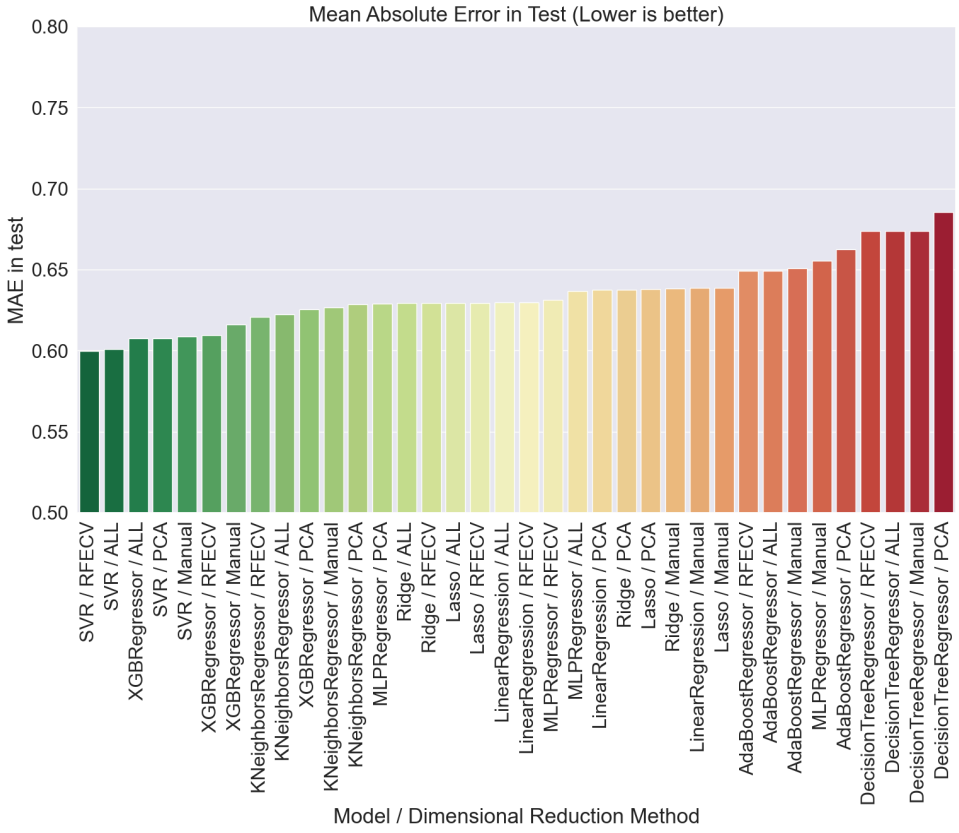
Model selection

For the model selection several methods were explored, firstly as the dataset contained a moderately high amount of predictors (45 numerical and 14 categorical dummies) different approaches of dimensionality reduction were examined aiming to generate simpler models easier to understand, improve accuracy and efficiency of the training process.
The following techniques were utilized:

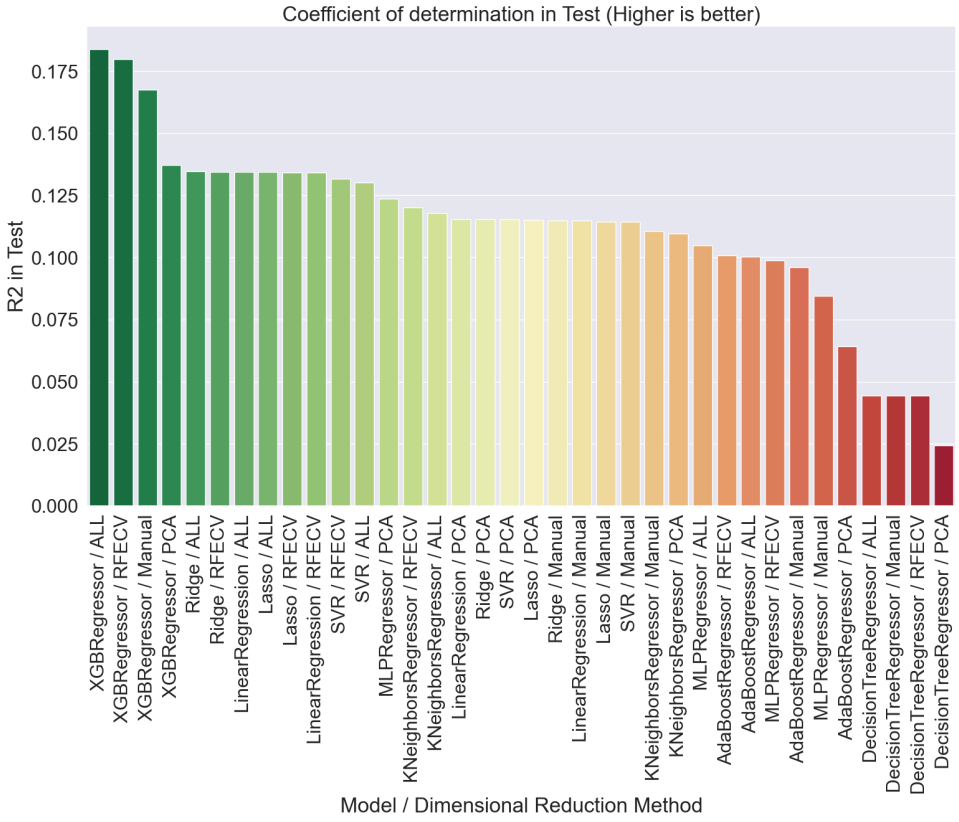
- Principal component analysis (PCA)
- Recursive Feature Elimination (RFECV)
- Manual feature selection

After each of the previous methods, several regressive models were explored using gridsearch, focusing on mean absolute error (**MAE**) as criterion as well as the coefficient of determination (**R2**), the models were also fitted with the complete 59 predictors without features selection (in the following figures the labels with the "ALL" tag, refer to those models running the whole set of predictors).





When comparing the MAE both in train and validation, the top performer models were support vector machine regression (SVR) and XGBoost implementation of gradient boosting, there was no clear improvement from any of the dimensionality reduction methods explored, with the model being the main factor influencing the error, by contrary the worst performing models were those paired with PCA.



For the final model selection, the most promising set of model and reduction techniques were further explored, tuning the hyperparameters and checking for robustness changing the train and test datasets.

While SVR models showed the best error in test dataset, they were overfitting in train, when explored for robustness SVR showed being sensible to changes in the train/test data split.

XGBoost performed more stable when changing the split, and also was the top performer in the coefficient of determination both in train and test, due to the previously mentioned XGBoost was selected as the final model paired with no data dimensionality reduction, as the speed for training this type of models is relatively fast .

The final parameters selected after the grid search for XGBoost were:

- learning_rate: 0.05
- max_depth: 4
- n_estimators: 300
- subsample: 0.66

The final model performance with a fixed random state of 42 had a MAE in train 0.566 (2271) and a MAE in test 0.608 (2293), for the coefficient of determination the results were R2 in train 0.332 and a R2 in test 0.184.

Conclusions

- Dimensionality reduction techniques helped reduce training time of some models but didn't show a clear improvement in objective functions or coefficient of determination.
- The coefficient of determination (R2) values for both train and test datasets were modest, suggesting that the model explains only a proportion of the variance in the target variable.
- There was no clear or high correlation between the predictors and the target variable, which indicates that this type of data is not sufficient or adequate for a highly accurate model for predicting likes in a particular review, further data such as more specific brand or model of the product can be interesting to explore.
- Collect more granular data related to the specific brand or model of the product being reviewed. This could provide additional insights into customer preferences and improve the predictive accuracy of the model.
- Conduct longitudinal analysis to observe how user preferences and behavior change over time, which could provide valuable insights for developing more accurate predictive models.