Kaggle competition

Author: Dorian Beganovic

Description of the model

1) FEATURE SELECTION

In the final model we used all original features unscaled and did not do any particular procedure to select some or process features. This decision was made as during testing we tried different methods of scaling features (MinMaxScaler, StandardScaler, RobustScaler) and found no improvement with those. Furthermore we tried to reduce the number of features using PCA but that also did not yield an improvement.

We transformed the target variable (shares) into log space to improve the performance of our model and we noticed it had a positive effect. Conversely for final prediction we transformed the predicted values back into normal range.

We dealt with missing values by replacing them with 0. There weren't any categorical features so we didn't have to deal with one hot encoding and likely reducing the dimensionality of data.

2) TRAINING

We performed 5-fold Cross validation on different models we tested (overall 25+ models). Cross-validation was very time consuming on our final model which averaged out predictions of 38 smaller linear models.

3) FINAL MODEL

We tried various linear models (gradient boosted regressors/xgboost, lasso, ridge, linear model, elastic net…) but none of those by itself produced good results as they usually were ranked in the bottom 15-20% on the leaderboard.

The final model we decided to worked by averaging the predicted values of various underlying linear models. We decided for this approach as classifiers like Random Forest prove that combining many smaller classifier and then combining their votes often yields better results than just using a single classifier.

In our final model we used 38 regressors and predicted value of each of those had equal weight in the final output.
Specifically the models were:

- 20 variations of gradient boosting regressor (similar to xgboost) which provides overall very good tradeoff between bias and variance because it's built from an ensemble of weak prediction models
- 5 Lasso regressors with different alpha values
- 5 Ridge regressors with different alpha values
- 5 Elastic net regressors with different alpha values
- 3 Simple Linear regressors

We used Lasso, Ridge and Elastic as they are known for not overfitting the data and providing a very good baseline model on unseen data.

We used python and libraries scikit-learn, numpy and pandas to make our predictions.