PREDICTING STOCK MOVEMENTS USING NEWS AND MARKET DATA
HELE-LIIS PEEDESK, VILLEM TÕNISSON, KÄTRIN SUVI

• INTRODUCTION
The aim of the project was to make predictions of stock movements. We made predictions based on news articles that contain information about certain assets, and based on previous returns and trading volumes.
The goal was to find a predictive model to predict how confident we are in positive or negative return of an asset based on news published 10 days earlier.

• DATA
We entered Kaggle competition and used the data provided in that competition.
Data about news articles covers the size and type of the article, for example the length of it, the first occurrence of related word (asset). It also contains information about the sentiment of the news article, i.e. whether the article had positive or negative sentiment towards the asset. This dataset contained multiple rows per asset and date, and needed to be aggregated.
The market dataset contains asset prices, returns and trading volumes in time-series. This dataset needed some cleaning, as there were erroneous return values, and typos in asset codes.

• METHODS USED
There were 3 methods mainly used: random forests, LGB and neural networks. All different methods were used with different approach on datasets. Additionally SVM and XGB methods were tried. LGBClassifier methods were tried on classified data.

• EVALUATION
The competition in Kaggle evaluates submissions by defined score. For each day \( t \) in evaluation period a size \( x_t \) was calculated, \( x_t = \sum_i y_{it} r_{it} u_{it} \), where \( y_{it} \) is a predicted value, \( r_{it} \) the return of the asset and \( u_{it} \) an indicator whether the asset is included to scoring on that day. The submission score was defined as

\[
\text{score} = \frac{x_t}{\sigma(x_t)}
\]
The same score was used for evaluation of the models trained in this project.

• RESULTS

<table>
<thead>
<tr>
<th>Method</th>
<th>Best score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random forest</td>
<td>0.52355</td>
</tr>
<tr>
<td>Neural networks</td>
<td>0.63687</td>
</tr>
<tr>
<td>Light GB</td>
<td>0.60657</td>
</tr>
</tbody>
</table>

• KERNELS USED
- Base for the neural network on market data: https://www.kaggle.com/christofhenkel/market-data-nn-baseline
- Base for LGBClassifier, Light GB: https://www.kaggle.com/alexnmartinez/minor-parameter-tuning-for-lightgbm-scaling-boost

Predictions using RandomForestRegressor
For random forest methods merged news and market data were used. Furthermore, as none of the assets does not have media coverage on every day, two separate models were fitted. The rows without news information (i.e. no coverage of the asset) were modelled based only on the previous returns and trading volumes, whereas the rows with news information used all of available data.
The used method for modelling just market data was linear regression. Random forest was used for merged dataset. Different numbers of trees, used features when choosing the split, maximal tree depth were tried.

Predictions using Light Gradient Boosting method
For light gradient boosting methods merged news and market data were used.
Light GBM is a gradient boosting framework that uses tree based learning algorithm. It grows tree vertically while other algorithm grows trees horizontally (grows tree leaf-wise). Model was trained using regression objective.
Different number of leafs, maximal depth, bagging fractions and frequensis were tried.

Predictions using NN and market data
Neural network models were constructed using only the historical market data.
Neural network was constructed by creating 2 different models, one for categorical values and the other for numerical values and then concatenating them.
Both models were quite simple, used only dense and batchnormalization layers. Keras module was used for creating the neural network. Rectified Linear Unit activation was used in layers where applicable and Adam optimizer was used in the concatenated model.

Categorized values were given unique id’s because the values needed to be numerical. The numerical values were standardized using StandardScaler.
The graph on the right shows the density of predicted confidence values on the test set.