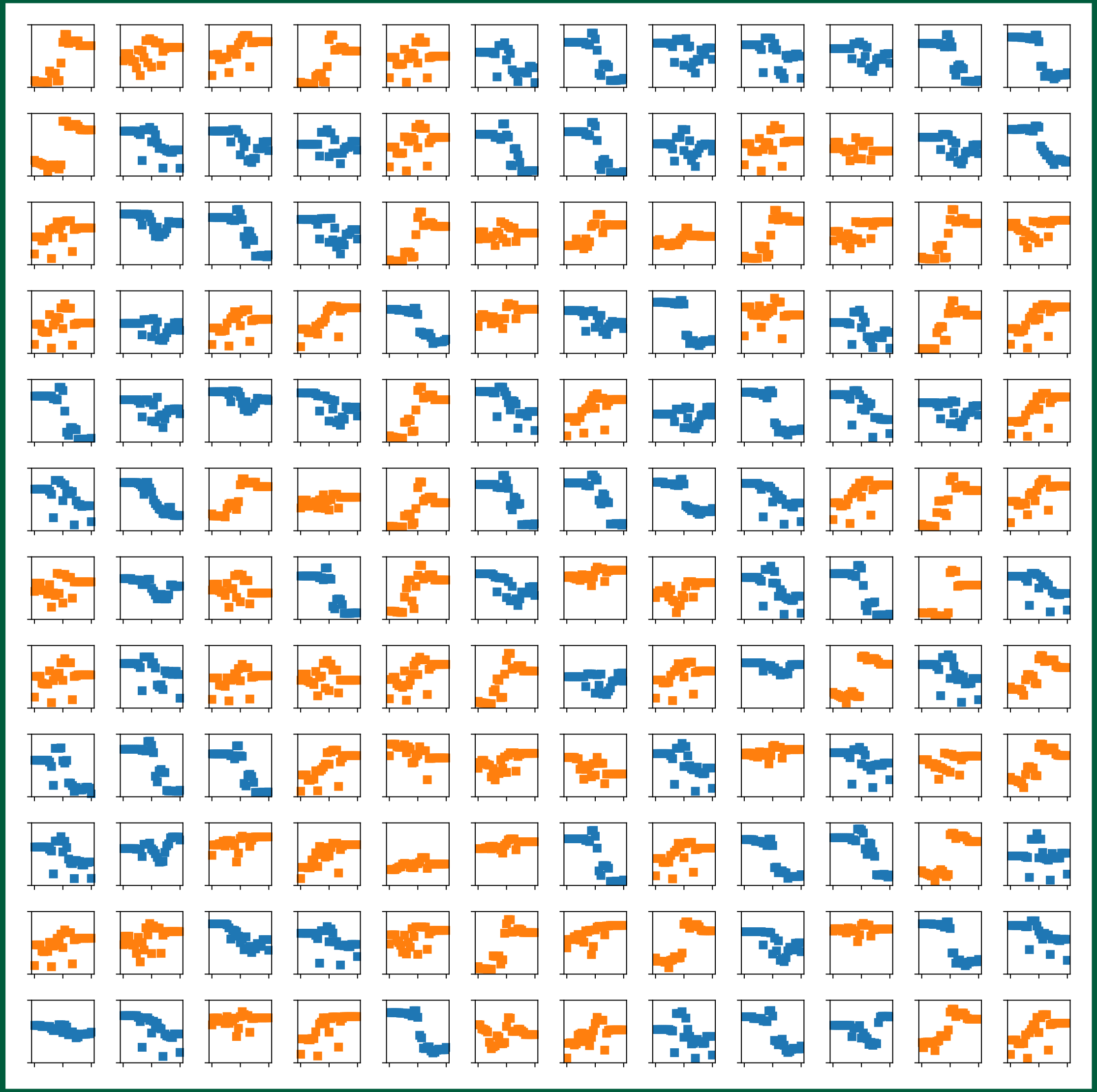


Market Making with ML

Outline

- Motivation
- Data
- Objectives
- ML
- Conclusions

The Danish Bond Market

What is traded?

Danish Mortgage-backed Bonds
Sizes of 1-500 Mio. DKK.
2500 different Bonds

Who is trading?

Buy side: Pension Funds, Hedge Funds(100+)
Sell side: Banks and Mortgage Institutions(10-15)

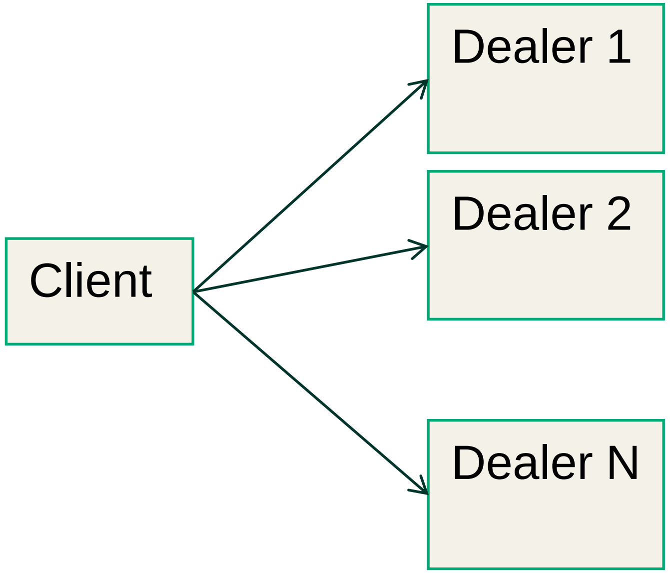
The Market is changing?

Moving from a traditional bilateral voice market to an electronic traded market using one trading platform

Motivation

- Electronic trading is possible
- The number of trades have gone up
- The dealers are getting fewer
- The number of different traded assets are growing
- The information related to each trade are growing
- The execution time for each trade is declining

Overview: Electronic trading



Process

- The Client sends a request to buy or sell an asset
- Each Dealer sends back a closed price, within 30 seconds
- The client chooses the best price and trades with one dealer

Points

- We are one of the dealers
- Other competitors are unknown to us
- If we lose, we do not know
  - Who won
  - The winning price

JYSKE BANK

Business objectives:

Assist a dealer in making faster and better decisions(auction bids).

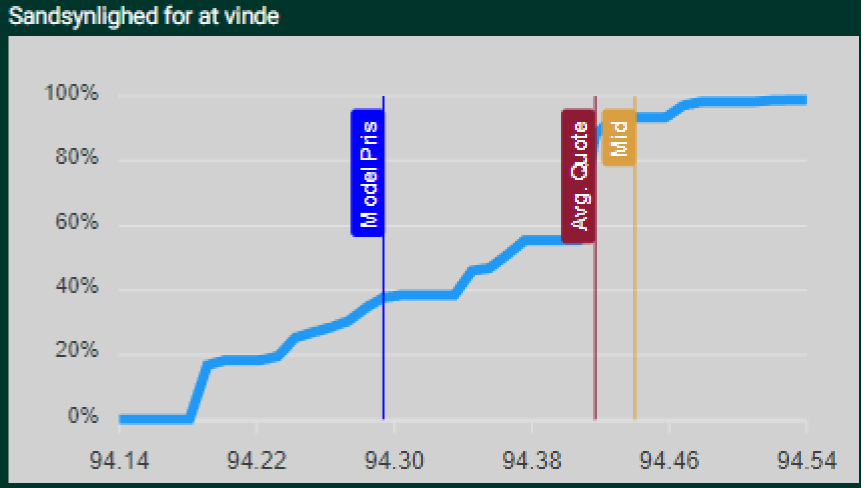
Improve on existing automatic “simple” rule-based trading systems.

Model Pris

94.295

(-0.146)

Navn	Værdi	Effekt ift. gns.
Bid-Ask Spread	0.4000	-0.0537 DKK
Oas	0.1149	-0.0253 DKK
Bpv	4.3556	-0.0144 DKK
Yield	2.2168	-0.0114 DKK
Dealers	5	+0.0086 DKK
RiskSpread	4.3556	-0.0078 DKK
Others		-0.0191 DKK



Available data

Highlight
- 12.000 Observations
- Win ratio is 20-30%
- Data categories
    - Auction data
    - Bond data
    - Bond risk measures
- Data is excel-like and have no gabs

Auction data	Bond data	Bond Risk Measures
Win/Second/Loss	Isin	Option Adjusted Spread
Buy/Sell	Issuer	Risk spread
Amount	Amortization	Vega
Client Id	Maturity	Duration
# in competition	Coupon	Convexity
Quoted Price	Type	
Traded Price		

JYSKE BANK

Data inspection

- Expected correlations
  - Low spread => high win ratio
  - High number of dealers => lower win ratio

- We see some degree of overlapping data between Win/Loss
  => It is not immediately obvious how to classify Win/Loss

JYSKE BANK

Data preprocessing

Rescaling prices (feature engineering):

- We change the price data to values relative to the absolute price level
- The price level is given by our fair price (*FP*)
  => Difference from the fair price is relevant
  => *Spread* = *QP* – *FP*

Encoding

- Categorial data such as the client Id’s are *one-hot* encoded

Feature normalization

- Done to help our neural network model

Extra data points using SMOTE

- Small number of tail observations
- Only on training data!

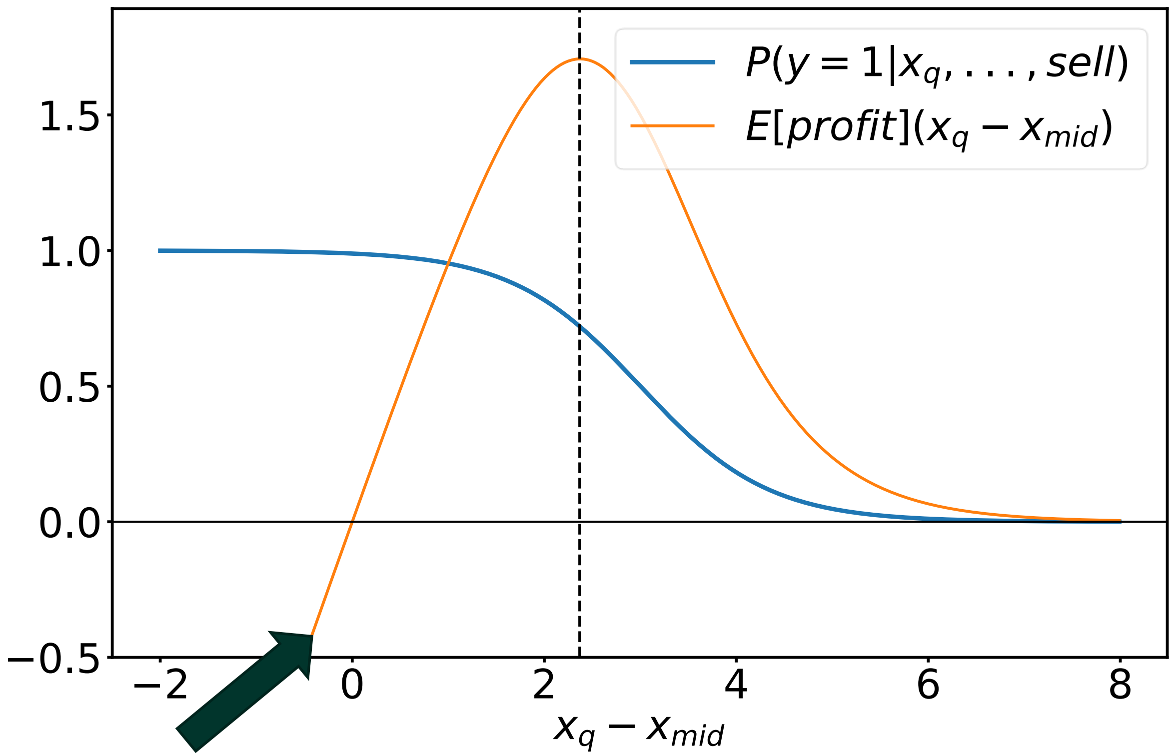
Split data on Buy/Sell

- Different distributions of spreads
- Buy and Sell act opposite to spread change

JYSKE BANK

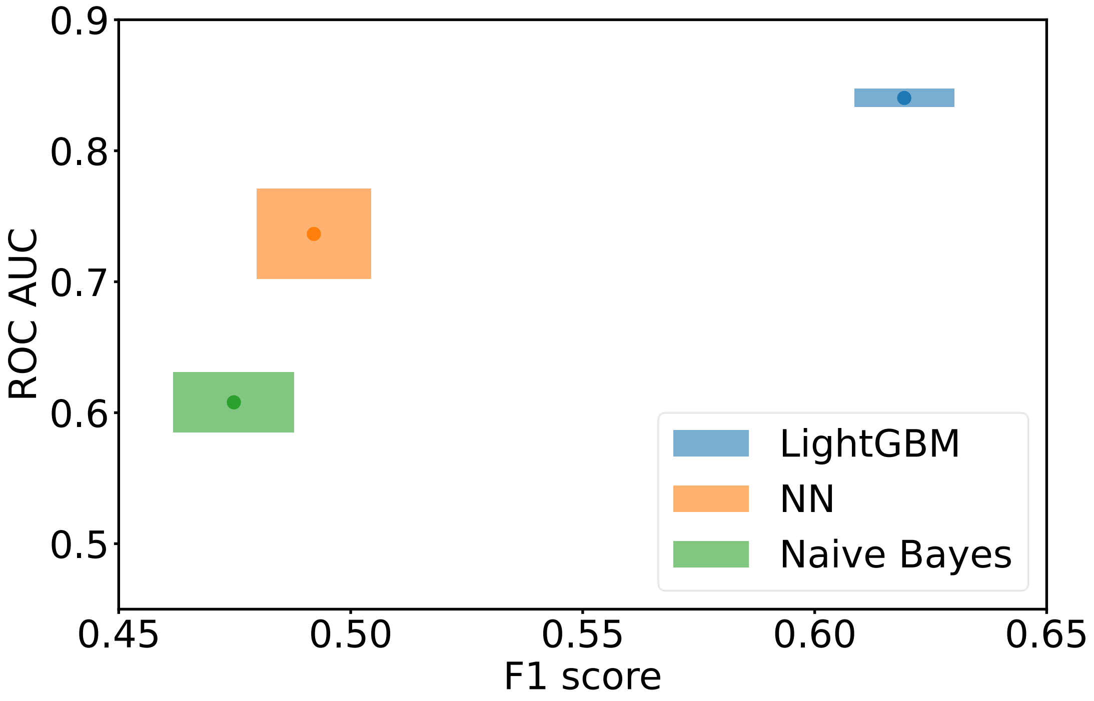
ML objectives:

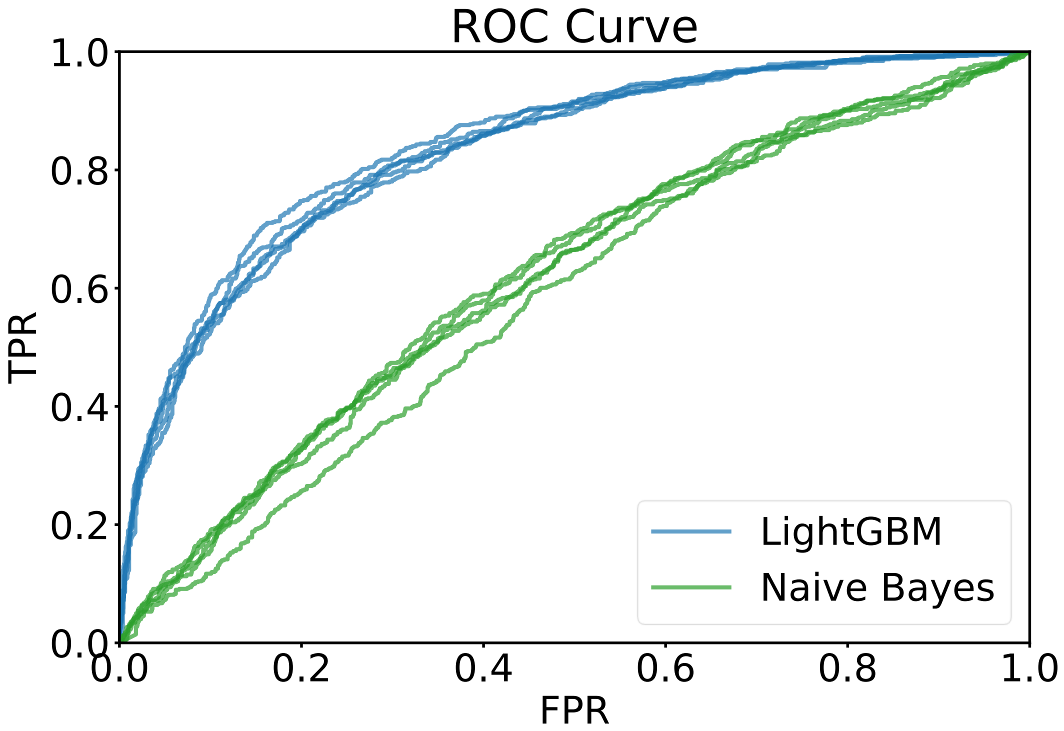
- Binary classification (Win/Lose)
- Predict the probability of winning an auction as a function of a quoted price x_q.
- Maximize the expected profit...



*E[profit] ~ P(y=1|x_q,...) * spread*

Choosing a model

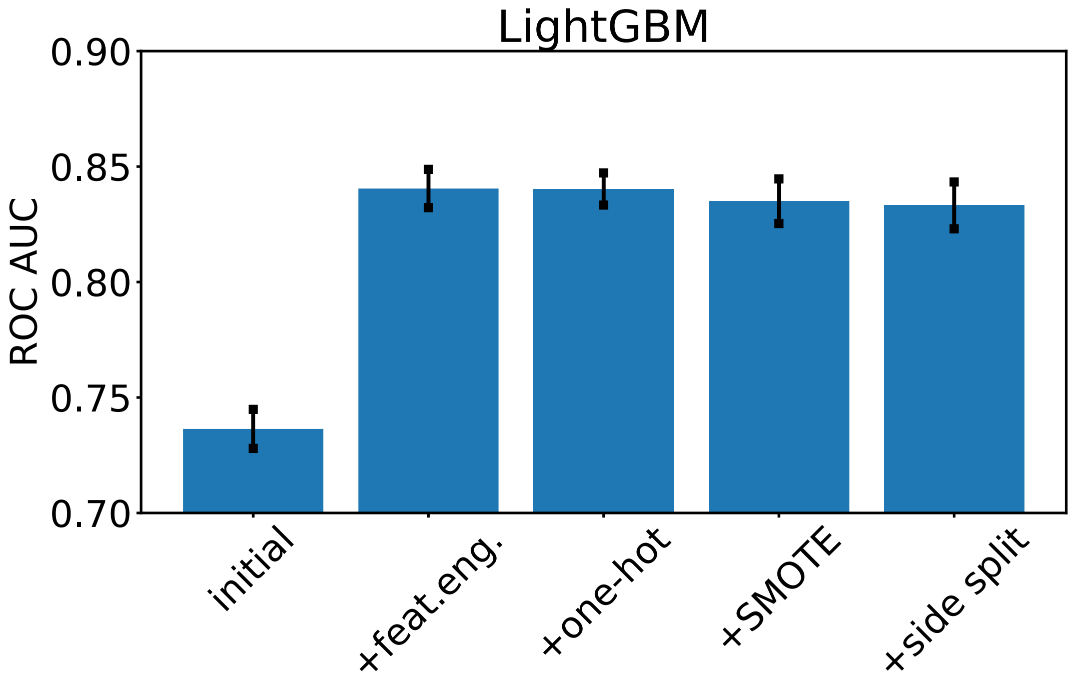


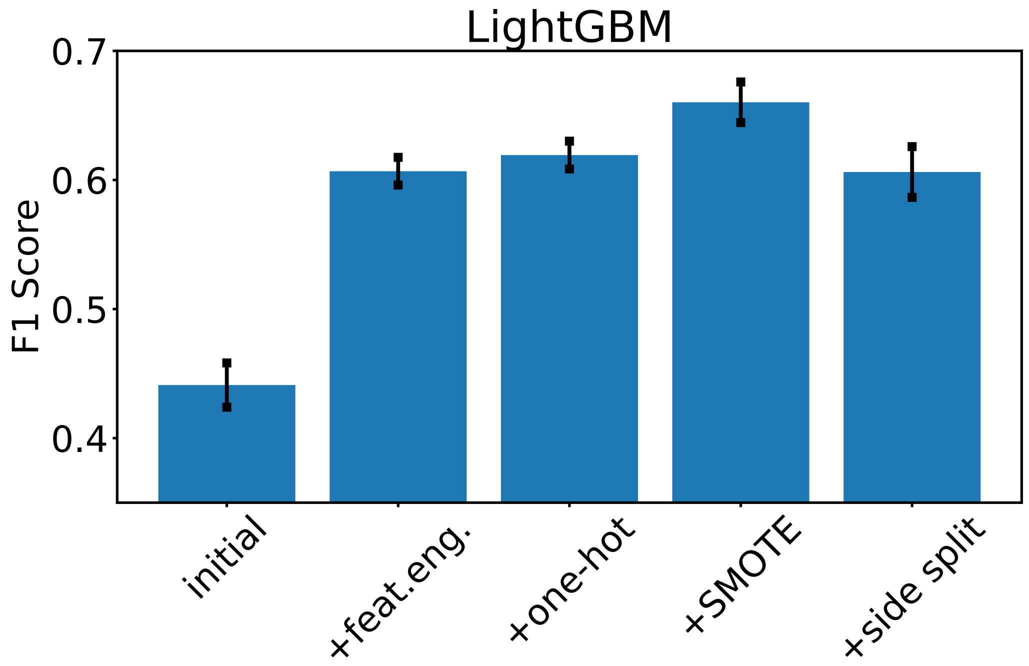


Center: mean
Box-size: std
From 5 fold cross-validation

LightGBM was chosen for the accuracy in both ROC AUC and F1 scores

Steps towards win/lose classification

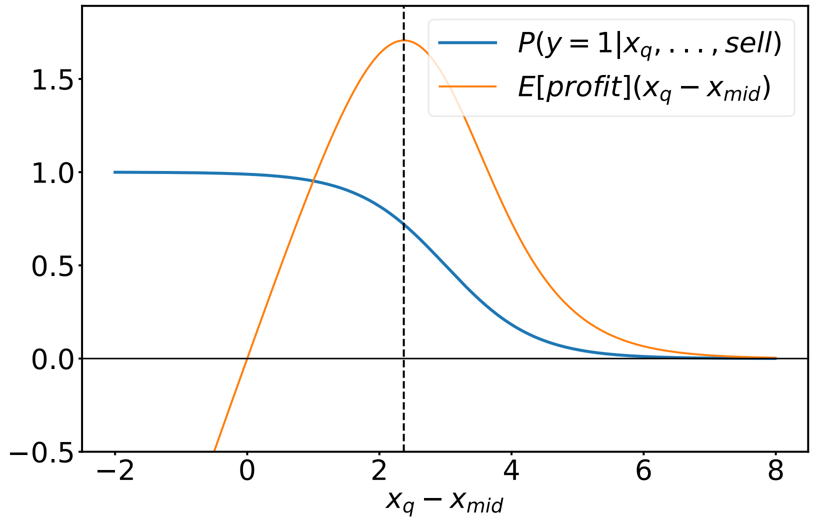


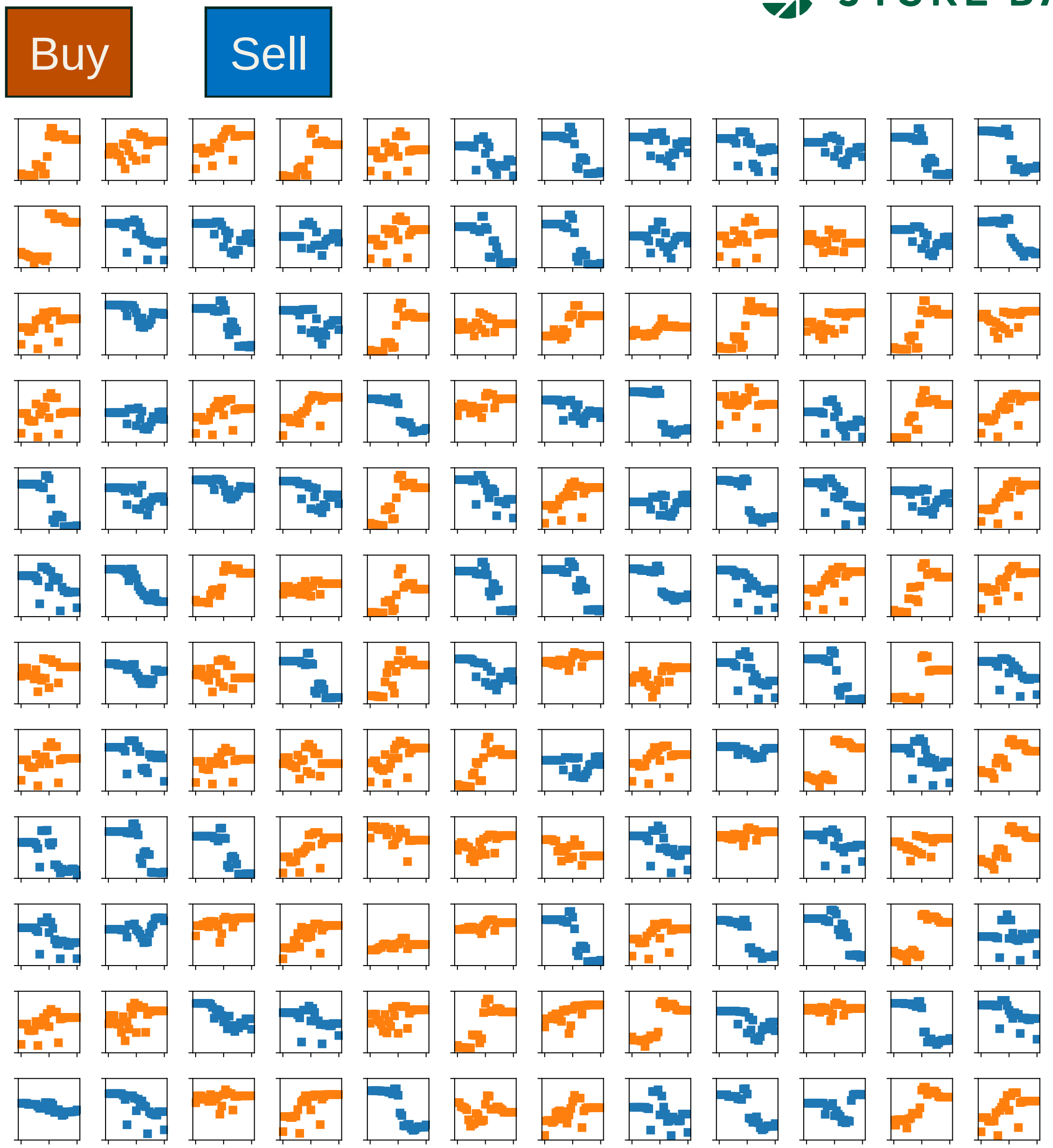


Introducing relative prices (feature engineering) is the most important step

Towards win probability as function of quoted price

$$P(y=1|(x_q, \ldots))$$

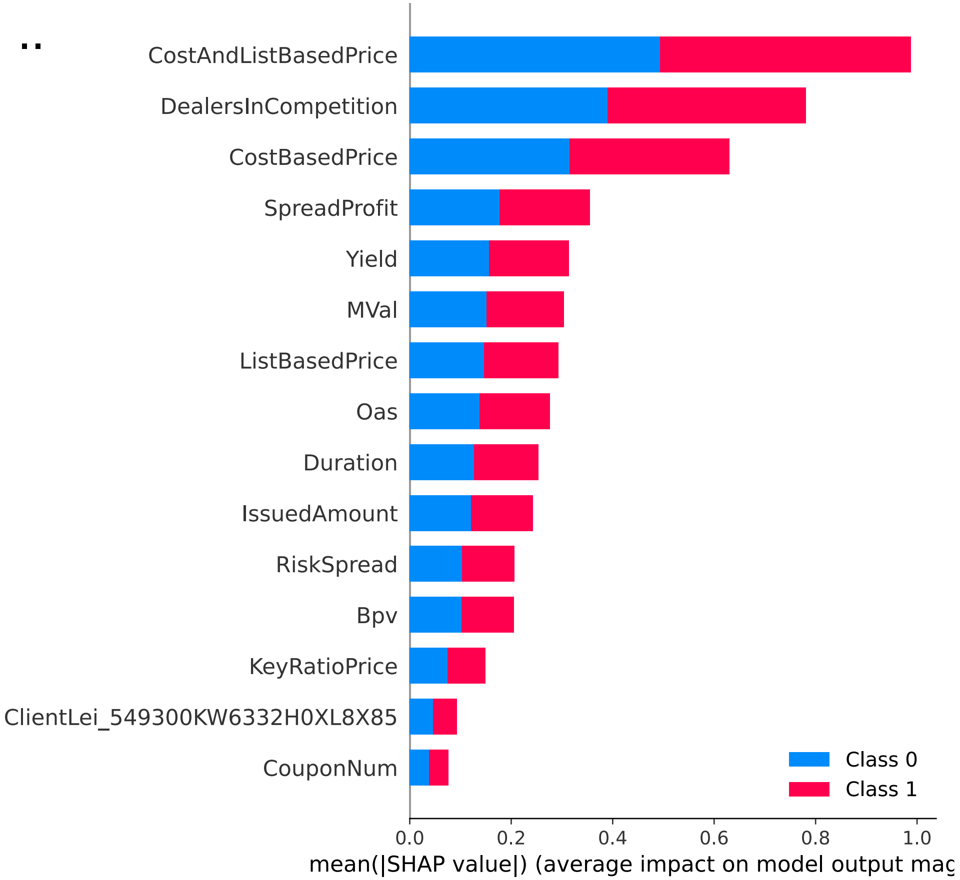


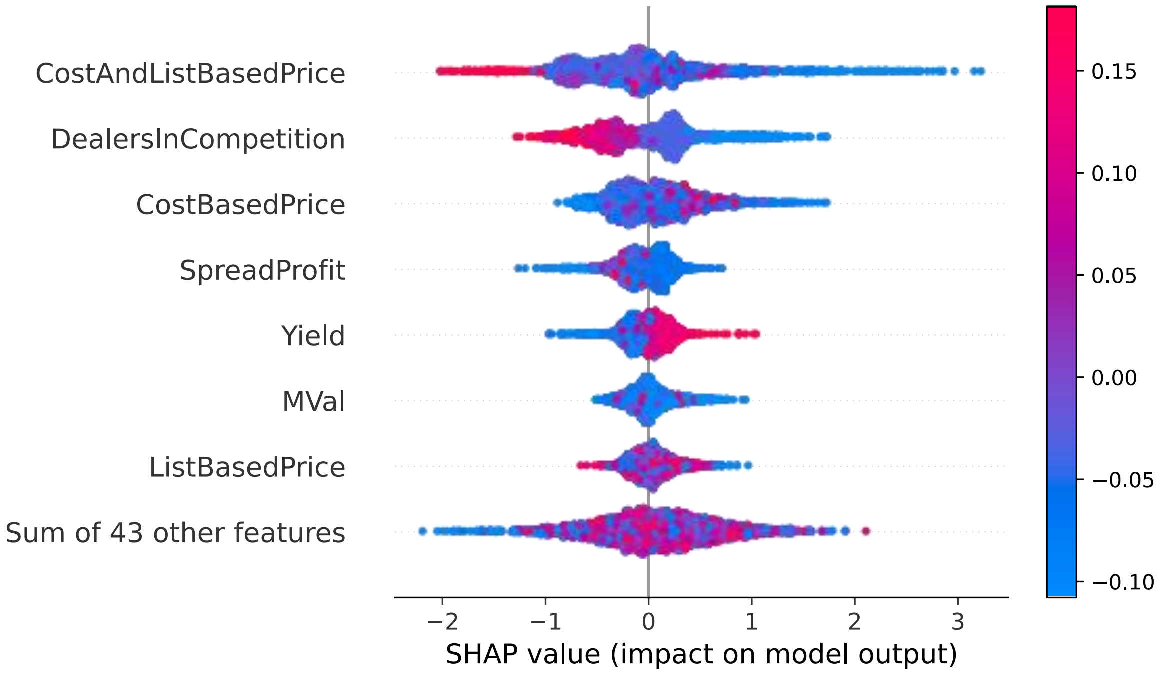


Trends are apparent … but more work is needed to make it useful in the real world…

Feature importance from SHAP

..

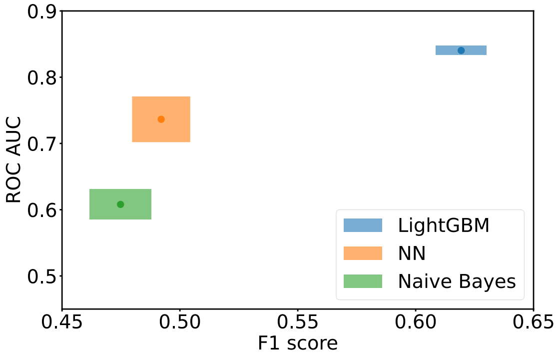




In general, feature ranking from SHAP correlates well with the dealers expectations.

JYSKE BANK

Conclusions



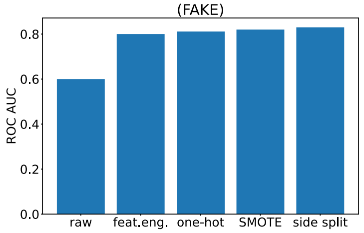
- LightGBM gave better results than NN and NB

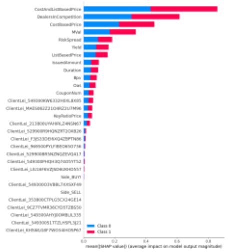
- Feature engineering (relative prices) important

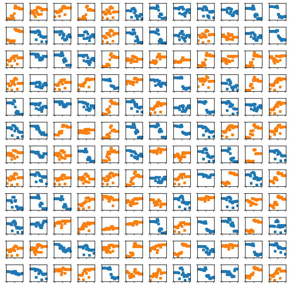
- Important features:
  - prices
  - number of competitors

- Trends are visible, but more work is needed to get something useful for the real world

- Proved that it should be possible to build a model which may support the dealers or, possibly, automate some trades







Future work

- Collect extra data (features)
- Estimate quote to maximize profit
- Compare with existing algo-trades and dealer-trades.
- Better fair price estimation

Thank you for the attention!

JYSKE BANK

Appendix

Feature transformation of prices

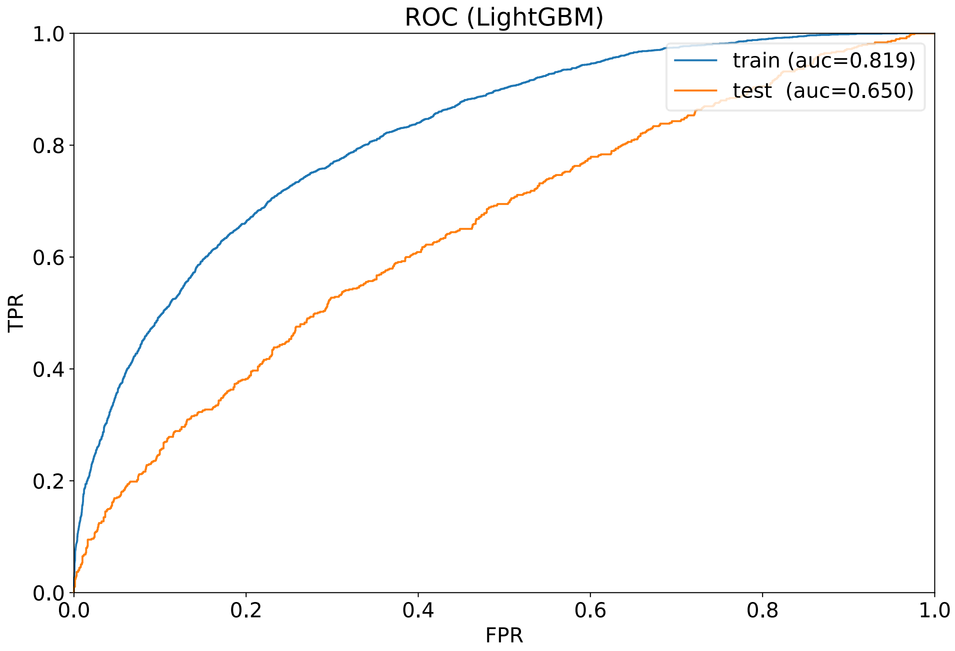
$$x_{q1} = \frac{x_q - x_c}{x_c} \qquad \text{(Cost-based price)}$$

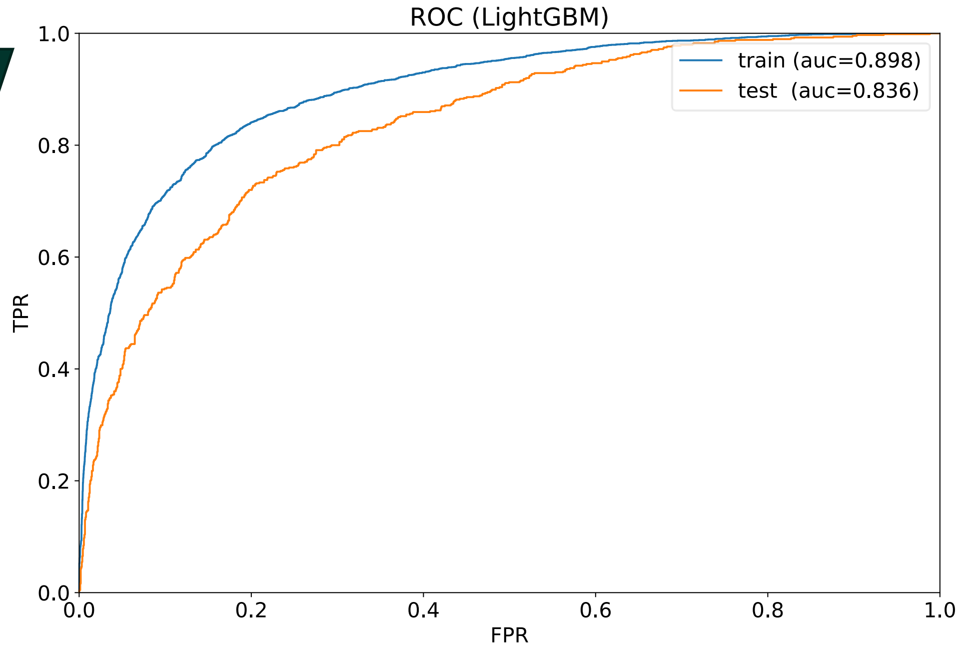
$$x_{q2} = \frac{x_q - x_c}{x_l - x_c} \qquad \text{(Cost- and List-based price)}$$

$$x_{q3} = \frac{x_q}{x_l} \qquad \text{(List-based price)}$$

Even a simple feature transform has a big effect

```python
def simple_feature_transform(row) -> float:
    sell_side = row.Side == "SELL"
    if sell_side:
        return row.QuotedPrice - row.Bid
    else:
        return -(row.QuotedPrice - row.Ask)
```





Result of Hyperparameter tuning for NN:

Model: Keras squential, dense layers (input: 49 neurons - "Relu", output: 1 neuron - "sigmoid"), binary crossentropy loss function

Tuned hyper parameters:
1) Number of hidden layers (1, …, 5)
2) Number of neurons per layer (49, 98, 147)
3) Activation functions in hidden layers (Relu, Sigmoid)
4) Learning rate (0,01, 0.01, 0.001)
5) Threshold in BinaryAccuracy-metric (0, …, 1 step 0.1)

Hyperparameter	Best value(s)
Number of layers	3
Layer 1	147 neurons, "Relu"
Layer 2	98 neurons, "Relu"
Layer 3	47 neurons, "Sigmoid"
Learning rate	0.01
Threshold	0.2

Result of Hyperparameter tuning for LightGBM

Model: LighGBM

```
best_params= {
    "learning_rate": 0.05,
    "max_depth":-1,
    "n_estimators": 200,
    "num_leaves": 60,
}
```