Machine Learning Methods for Vanilla Option Pricing and Heston Calibration
PRACTICUM 2023

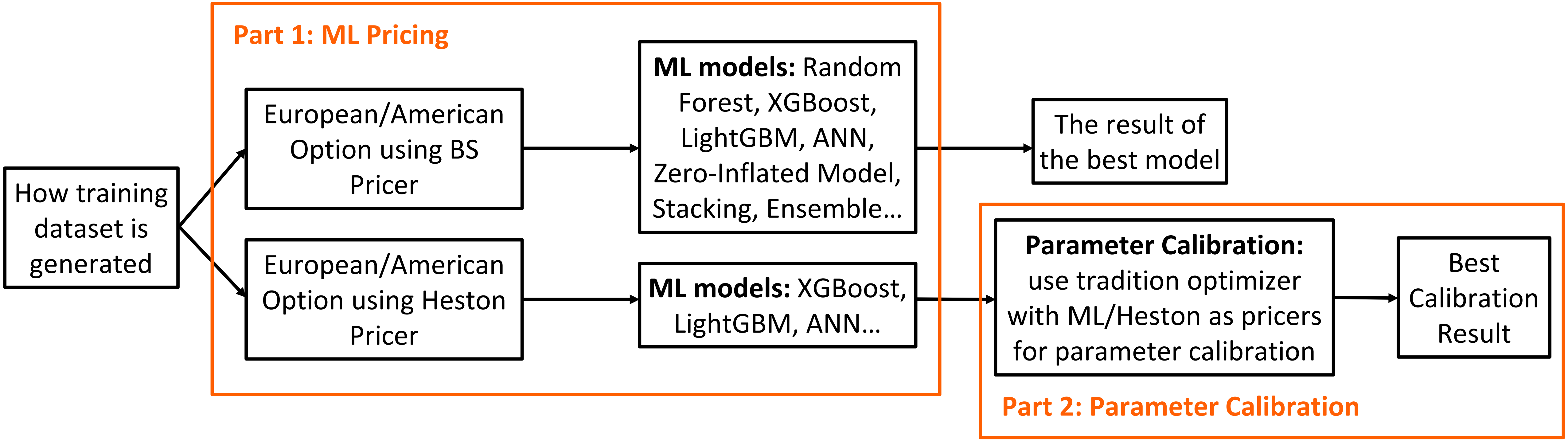
Team Members: Xiaoyue Chen, Chang Gao, Tiantian Hu, Jiangda Wang
Sponsor: Ilan Zlotin




Motivation

- **Stochastic volatility model**
  widely used but computationally expensive and difficult to calibrate
- **Machine learning methods**
  fast and accurate solutions to complex problems
- **Deep neural networks**
  providing a more efficient way to price options compared to traditional methods
- **Using machine learning methods for option pricing and calibration** can potentially lead to more accurate and efficient solutions, improving the overall performance of the option pricing models**.**

Outline



Part 1: ML Pricing
How training dataset is generated
European/American Option using BS Pricer
European/American Option using Heston Pricer
ML models: Random Forest, XGBoost, LightGBM, ANN, Zero-Inflated Model, Stacking, Ensemble…
ML models: XGBoost, LightGBM, ANN…
The result of the best model
Parameter Calibration: use tradition optimizer with ML/Heston as pricers for parameter calibration
Best Calibration Result
Part 2: Parameter Calibration

Datasets

Pricing option with Machine Learning:

1. European option
   Black-Scholes Formula
2. European option
   Heston with Fourier Transform Method
3. American option
   Crank-Nicolson Finite Differences
4. American option
   Heston with PDE Method

Heston Calibration:

5. European option (5k simulated market price)
   Heston with Fourier Transform Method
6. American option (5k simulated market price)
   Heston with PDE Method

Notation

Market Parameters

- $\boldsymbol{m}$ $(\frac{K}{S_0})$: moneyness
- $\boldsymbol{T}$: time to maturity
- $\boldsymbol{q}$: dividend rate
- $\boldsymbol{r}$: riskless interest rate

Heston Parameters

- $\boldsymbol{v_0}$: initial volatility
- $\boldsymbol{\theta}$: long-term mean of the stochastic volatility
- $\boldsymbol{\kappa}$: rate at which the stochastic volatility reverts to its long-term mean, θ
- $\boldsymbol{\sigma}$: volatility of the volatility
- $\boldsymbol{\rho}$: randomness in the asset price and volatility

Datasets

Set 1&3:

BSM & CNFD

- $m \in [0.2, 5]$
- $T \in [0.004, 2]$
- $\sigma \in [0.05, 2]$
- $q \in [0, 0.05]$
- $r \in [0, 0.08]$

Set 2&4:

Heston with FT & PDE

Market Parameters

- $m \in [0.2, 5]$
- $T \in [0.004, 2]$
- $q \in [0, 0.05]$
- $r \in [0, 0.08]$

Heston Parameters

- $v_0 \in [0.01, 0.5]$
- $\theta \in [0.01, 2.0]$
- $\kappa \in [0.01, 2{,}0]$
- $\sigma \in [0.01, 1.0]$
- $\rho \in [-0.9, 0.9]$

Set 5&6:

Heston Calibration (3 sets)

Market Parameters

- $m \in [0.2, 5]$
- $T \in [0.004, 2]$
- $q = 0$

Heston Parameters and r

$$[r, v_0, \theta, \kappa, \sigma, \rho]$$

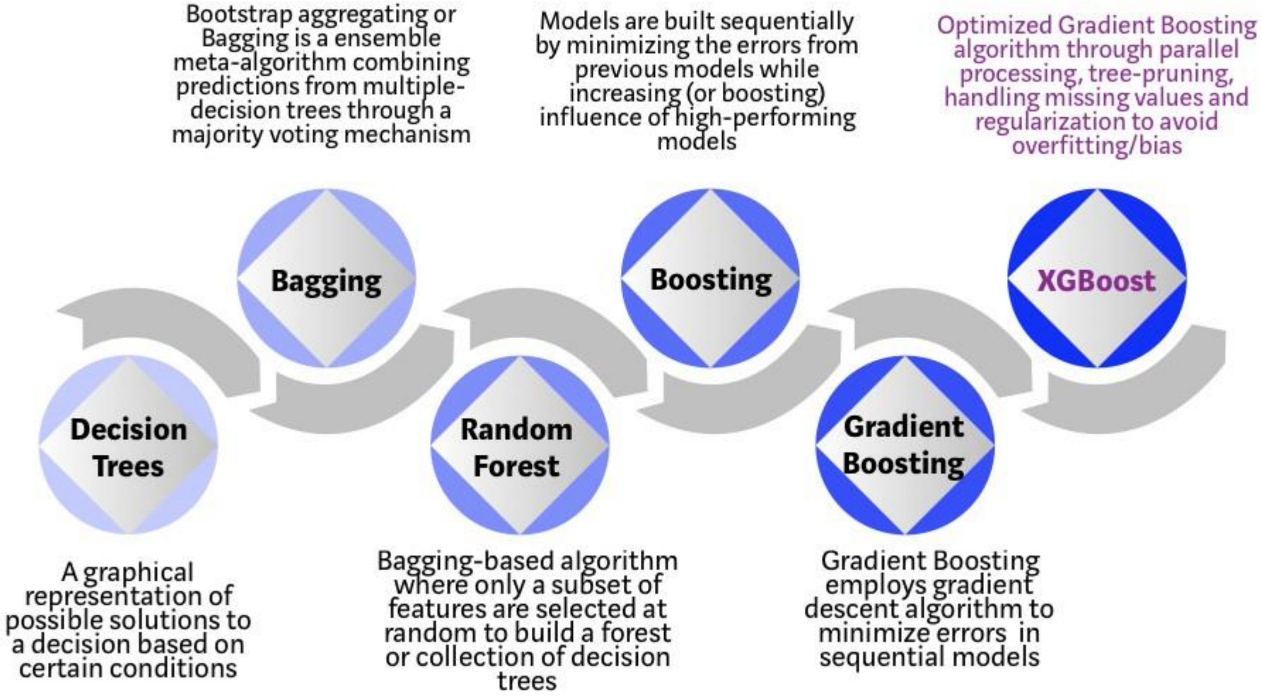
- $[0.02\ , 0.04, 0.04, 1.5, 0.3, -0.5]$
- $[0.03, 0.01, 0.02, 1.\ , 0.1, 0.2]$
- $[0.05, 0.06, 0.06, 1.5, 0.3, -0.6]$

ML Pricing - European/American Option

Procedure

1. **Generate the data set for training the Machine Learning models**
    - BSM: Using the BS pricer methodology for generating the European option and the Crank-Nicolson finite difference method for generating the American option
    - Heston: Using the Heston Fourier pricer for generating the European option and Heston PDE method for generating the American option
2. **Utilize machine learning models for pricing European and American options**
    - Apply the **Random Forest**, **XGBoost**, **CatBoost, LightGBM**, and **ANN** machine learning algorithms for pricing European and American options.
3. **Compare the performance of different machine learning methods to identify the optimal model**

ML Pricing - European/American Option
Model Introduction- XGBoost



Key Features:

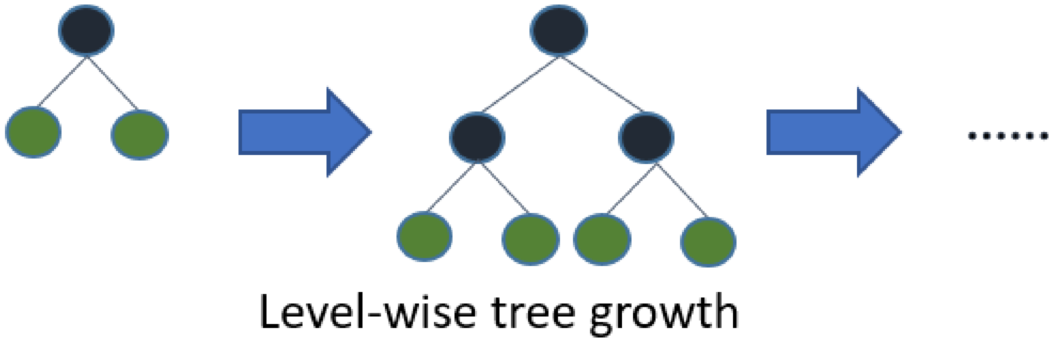
- a supervised learning algorithm based on ensemble learning.
- Using gradient boosting algorithm that builds trees one at a time. (level-wise growth)
- treats continuous values by calculating the best split point based on the data's distribution.
- Handle missing values more decently.

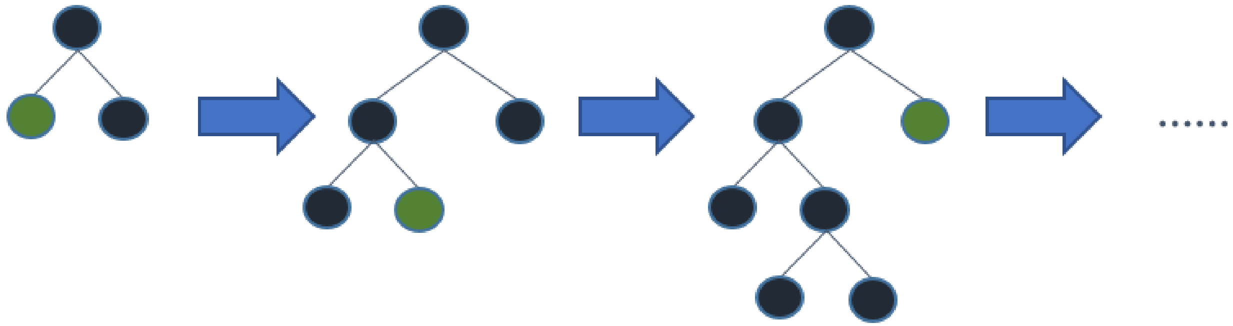
Important Hyperparameters

- ***max_depth***: Maximum depth of a tree
- ***min_child_weight***: Minimum sum of instance weight (hessian) needed in a child
- ***colsample_bytree***: specify the fraction of columns to be subsampled when constructing a tree
- ***reg_lambda***: L2 regularization term on weights.
- ***reg_alpha***: L1 regularization term on weights.

ML Pricing - European/American Option

Model Introduction-LightGBM





Difference to XGBoost Model:

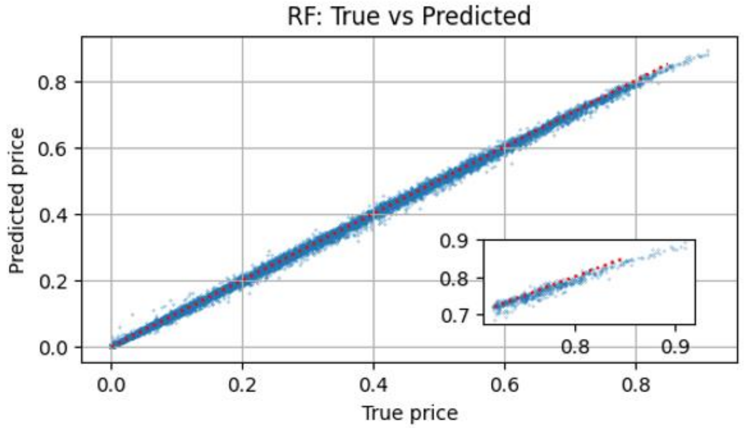
- LightGBM use a histogram-based algorithm that performs bucketing of values
- Leaf-wise (vertical) growth vs level-wise(horizontal) growth

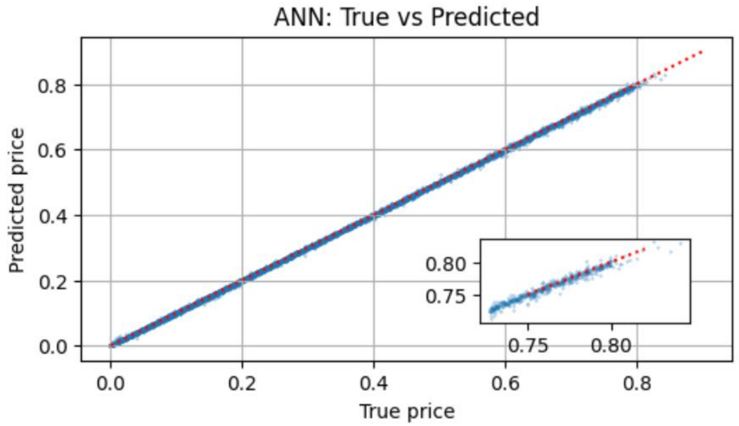
Important Hyper Parameters:

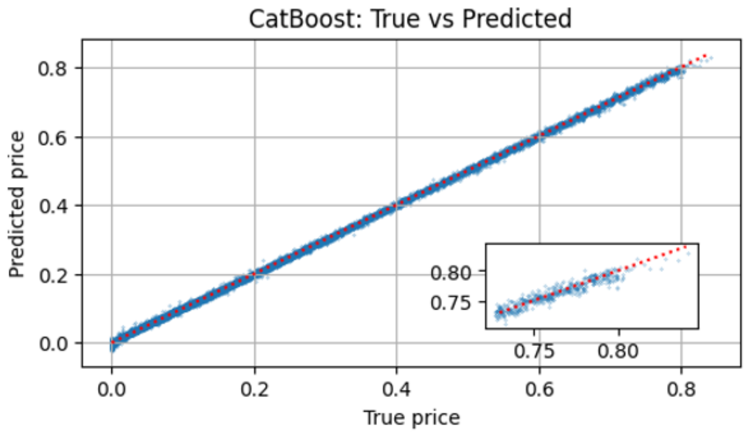
- ***num_leaves***: max number of leaves in one tree
- ***min_data_in_leaf*** **:** minimal number of data in one leaf.
- ***max_bin***: max number of bins that feature values will be bucketed in
- ***max_depth*****:** limit the max depth for tree mode
- ***lambda_l1, lambda_l2*****:** L1 & L2 regularization

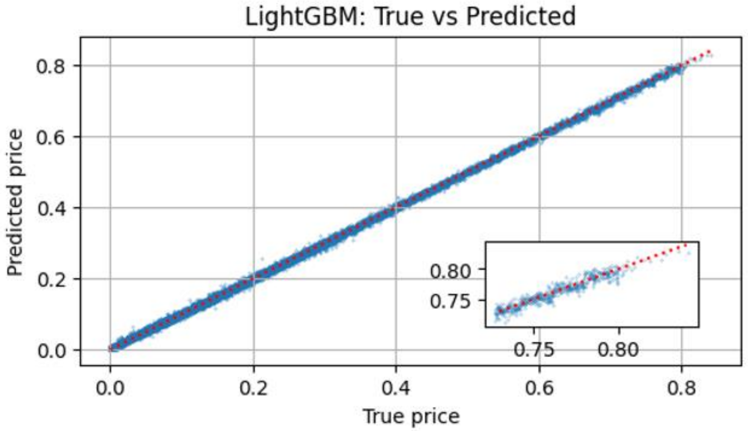
ML Pricing - European/American Option

True price vs Predicted price





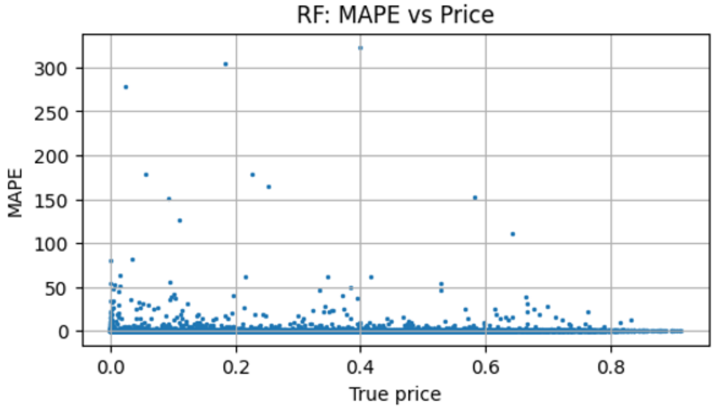


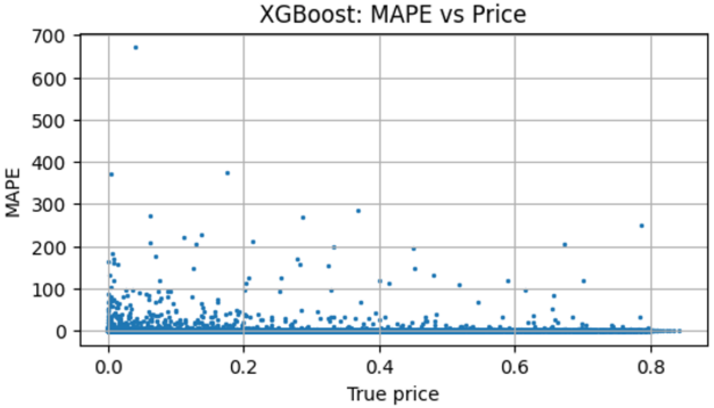


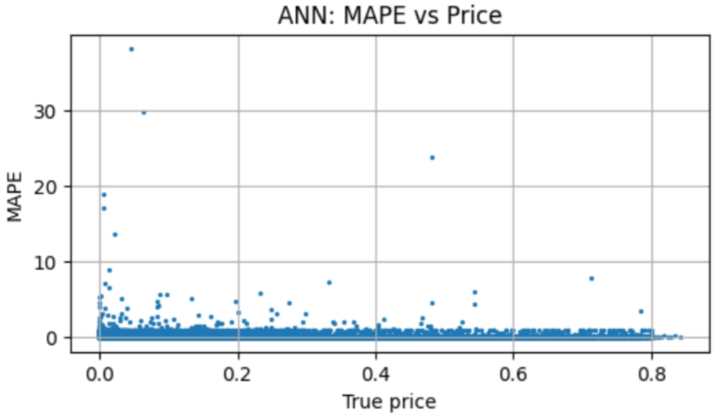
- The predicted price can be better fitted by all four machine learning methods.
- All models exhibit a proclivity for underestimating true prices as the price of underlying asset increases.

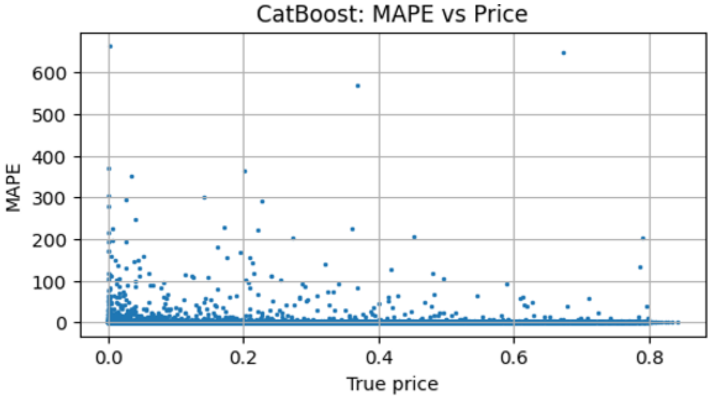
ML Pricing - European/American Option

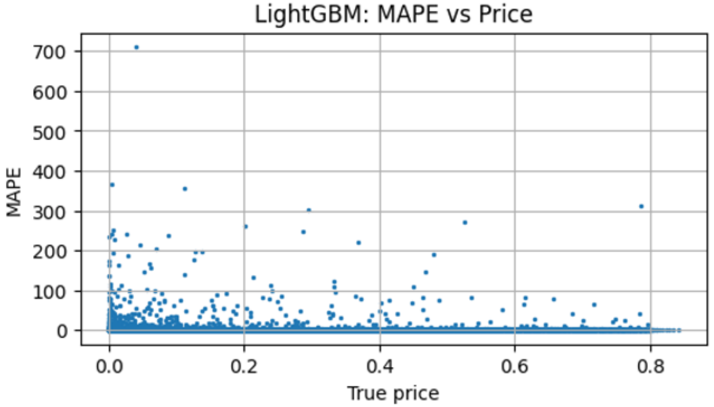
MAPE vs Price











- The Mean Absolute Percentage Error (MAPE) exhibits a tendency towards larger values as the true price converges to zero.

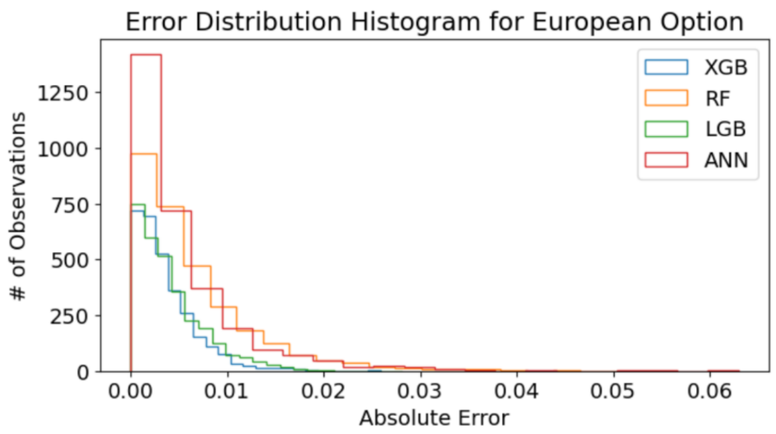
ML Pricing - European/American Option on BS Model

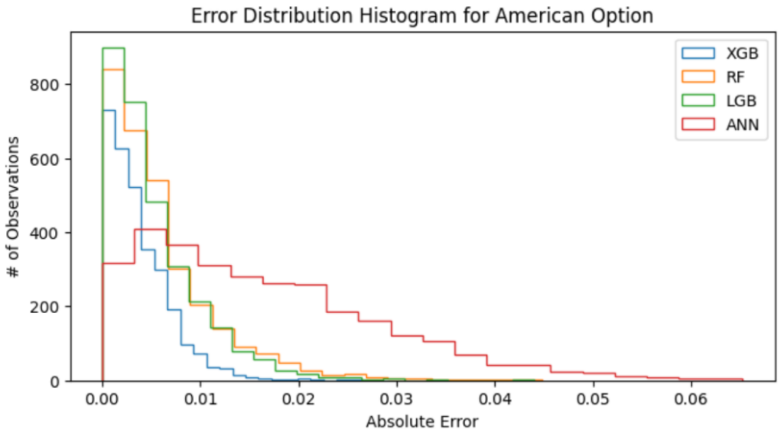
Model Comparison

- Three machine learning models are applied to option pricing, whereby **MAE, RMSE**, and **feature importance** are compared to determine the optimal model.
- **Performance:** XGBoost > LightGBM > ANN > RF

	American Call Option		European Call Option	
	RMSE	MAE	RMSE	MAE
XGBoost	0.00521	0.0039	0.0049	0.00379
LightGBM	0.00576	0.0043	0.0061	0.00463
ANN	0.00817	0.0055	0.0084	0.00558
RF	0.00815	0.0059	0.0091	0.00626

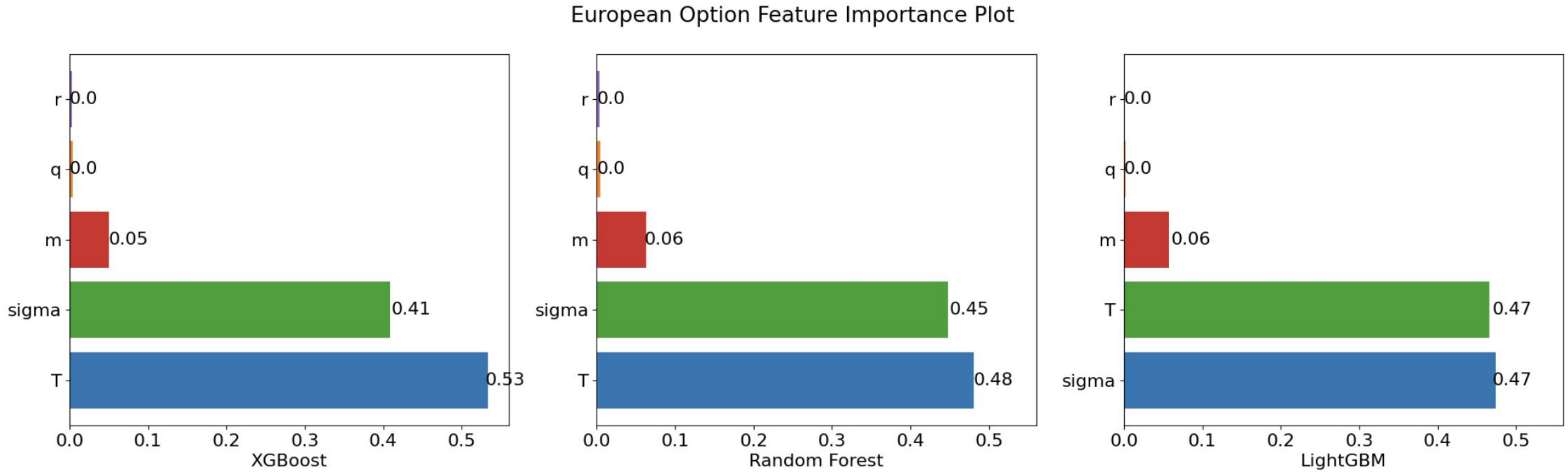
We pay attention to the absolute error quantile. XGBoost demonstrates the best performance in controlling absolute error.





ML Pricing - European/American Option on BS Model

Feature Importance



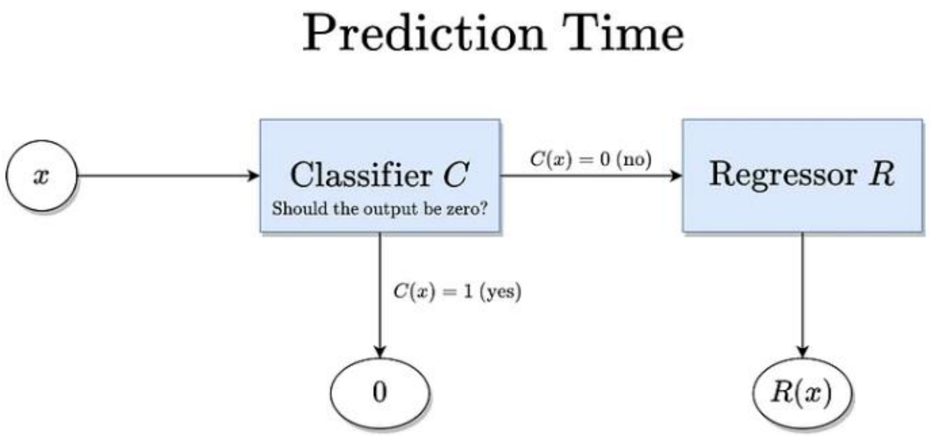
- The distributions of feature importance between European and American options in three ML methods are fundamentally similar.
- Three of the models exhibit a similar feature importance pattern to Volatility and Time to Maturity being the most significant factor.

ML Pricing - European/American Option on BS Model

Extended Analysis: Zero Inflated Problem

- Upon examining the Absolute Percentage Error in Random Forest, we discovered that the probability of the RF model producing a zero output is relatively small.

```
RF Absolute Percentage Error (quantile):
0.00     0.000
0.10     0.004
0.25     0.012
0.50     0.030
0.75     0.106
0.90     1.853
0.99       NaN
1.00       NaN
Name: g_eurocall, dtype: float64
```



1. Train a **Classifier C** that identify whether the regression output is 0
2. Train a **regressor R** on the part of the data with a non-zero target.

ML Pricing - European/American Option on BS Model

Extended Analysis: Zero Inflated Problem

Random Forest

```
Current Best Random Forest Performance:
RandomForestRegressor(max_features=3, n_estimators=200)
Train Set rmse:  0.0028543358021890825
Random Forest rmse:  0.007738857112653168 mae: 0.004923425898517992
Random Forest r2:  0.9989264282351908
```

RF Classifier + Regressor

```
Current Best RF Combined Performance:
ZeroInflatedRegressor(classifier=RandomForestClassifier(max_features=3,
                                                        n_estimators=200),
                      regressor=RandomForestRegressor(max_features=3,
                                                      n_estimators=200))
Train Set rmse:  0.0029062471645867993
RF Combined rmse:  0.007674041366281163 mae: 0.004911076764332204
RF Combined r2:  0.9989443360365766
```

Regressor

```
RF Absolute Error (quantile):
0.00    0.000000
0.25    0.000572
0.50    0.002805
0.75    0.007181
0.85    0.010345
0.90    0.012780
0.99    0.026723
1.00    0.076111
Name: g_eurocall, dtype: float64

RF Absolute Percentage Error (quantile):
0.00    9.329797e-07
0.10    4.414145e-03
0.25    1.155709e-02
0.50    3.028926e-02
0.75    1.082960e-01
0.90    1.949813e+00
0.99             NaN
1.00             NaN
Name: g_eurocall, dtype: float64
```

Classifier + Regressor

```
RF Absolute Error (quantile):
0.00    0.000000
0.25    0.000583
0.50    0.002850
0.75    0.007080
0.85    0.010085
0.90    0.012403
0.99    0.025760
1.00    0.083385
Name: g_eurocall, dtype: float64

RF Absolute Percentage Error (quantile):
0.00     0.000005
0.10     0.004108
0.25     0.010572
0.50     0.026486
0.75     0.076299
0.90     0.399884
0.99    22.895421
1.00          NaN
Name: g_eurocall, dtype: float64
```

- The RMSE and MAE of the combined model demonstrate minor improvement
- By integrating the classifier and regressor, the absolute error (AE) observed between the 0.75 and 0.99 quantiles exhibits a decrease. However, there is a slight increase in the AE between the 0 and 0.5 quantiles.
- The APE of the combined model is significantly lower compared to that of the model only with a regressor.

ML Pricing - European/American Option on BS Model

Extended Analysis: Zero Inflated Problem

Best Classifier Models:

Model	Accuracy
XGBoost	0.996
XGBoost	0.994
LightGBM	0.994
LightGBM	0.9936
Random Forest	0.992

Absolute Error:

	XGB	LGB	Stack_XGB	Stack_XGB_LGB
0.00	0.000	0.000	0.0000	0.0000
0.25	0.001	0.001	0.0009	0.0008
0.5	0.003	0.003	0.0028	0.0026
0.75	0.006	0.007	0.005456	0.0059
0.9	0.028	0.012	0.007266	0.0090
0.99	0.018	0.018	0.016439	0.0180
1.00	0.103	0.035	0.035281	0.0351

Absolute percentage error:

	XGB	LGB	Stacked_XGB	Stacked_XGB_LGB
0	0.000	0.000	0.000	0.000
0.25	0.007	0.007	0.008	0.005
0.50	0.020	0.021	0.021	0.013
0.75	0.089	0.078	0.066	0.047
0.85	0.170	0.162	0.157	0.148
0.90	1.220	0.600	0.383	0.473
0.99	451.000	276.317	55.602	92.337
1	Nan	Nan	Nan	Nan

```
Current Best xgb classifier Performance:
XGBClassifier(lambda_l2=0, n_estimators=800, subsample=0.6)
Train Set rmse:  0.0
xgb classifier rmse:  0.0621825270205921 mae: 0.0038666666666666667
Accuracy 0.9961333333333333
```

- The accuracy of the best classifier is 0.996
- The stacked model performs slightly better than the original model, but the improvement is not significant

ML Pricing - European/American Option on BS Model

Extended Analysis: BS EuroCall as feature to train AmerCall

Dataset:

	g_m	g_T	g_r	g_q	g_sigma	g_eurocall	g_amercall_fde
21	1.056413	0.126027	0.032595	0.006992	1.010777	0.121045	0.121043
32	4.974514	1.199890	0.062143	0.003586	1.355464	0.208305	0.208303
60	2.335682	1.046575	0.031766	0.003246	0.919297	0.139497	0.139497
70	1.066268	0.049315	0.069473	0.030717	1.853887	0.137895	0.137893
93	1.329132	0.106849	0.021158	0.005569	1.098013	0.050243	0.050242
143	0.985998	0.451945	0.075035	0.009029	0.637038	0.186951	0.186950
144	1.114553	0.126027	0.069473	0.027804	1.235267	0.133121	0.133120
172	3.530173	0.528658	0.069152	0.008453	1.604309	0.143564	0.143563

Motivation:

- American Option is hard to price => require checking **early exercise** condition
- European Option are often easier to price => BS formula for BS model/Fourier Transformation for Heston model
- European Option price could give useful information most of time.(here refer to call price)

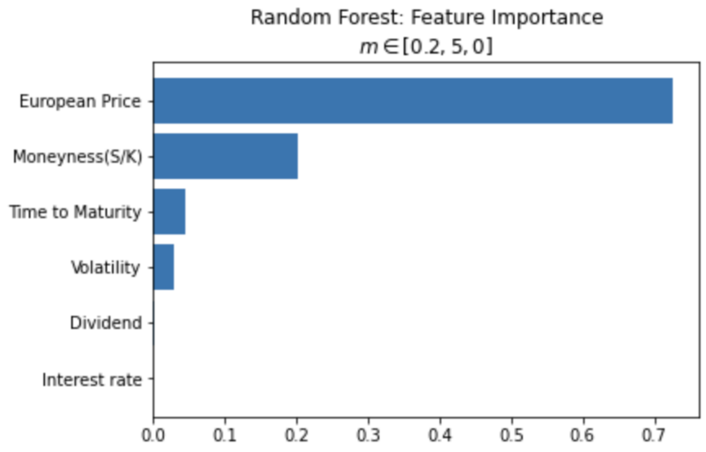
Observation:

We assume underlying asset to have price 1 and therefore in this case, the difference between American and European Call price is small, almost negligible.

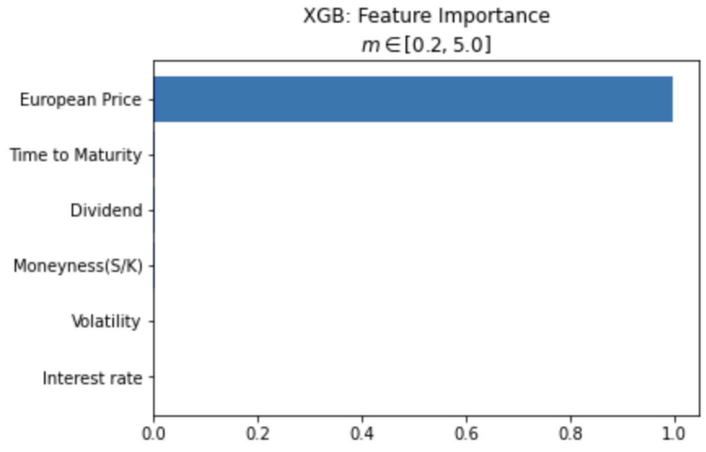
ML Pricing - European/American Option on BS Model

Extended Analysis: BS EuroCall as feature to train AmerCall

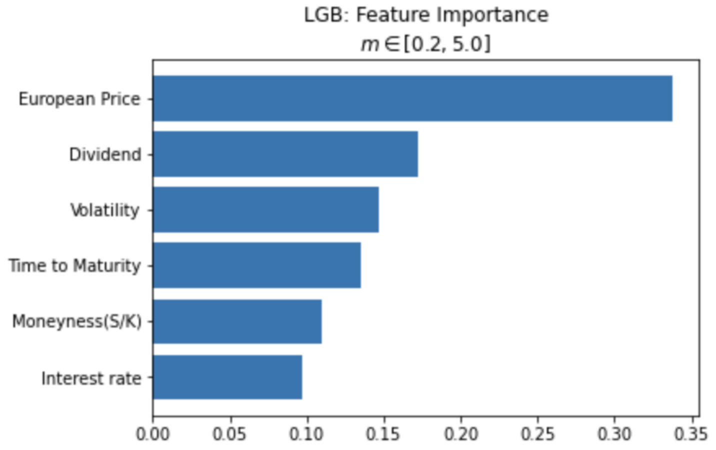
Random Forest



XGBoost



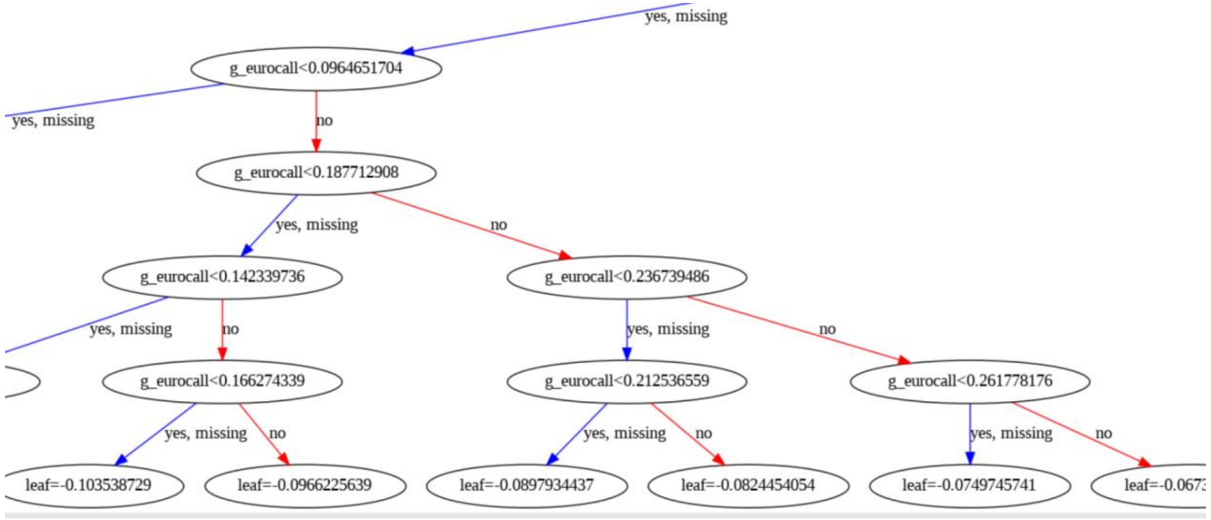
LightGBM



- European call option price plays an important part in all three models.
- The plot for LGBM is not correct, European Price should take about 80% - 100% of weight.
- The reason why XGBoost uses European option price purely might be XGBoost uses boosting method to train the model, which means it keeps improving the model based on the previous learning results, while RF just uses bagging to add randomness.

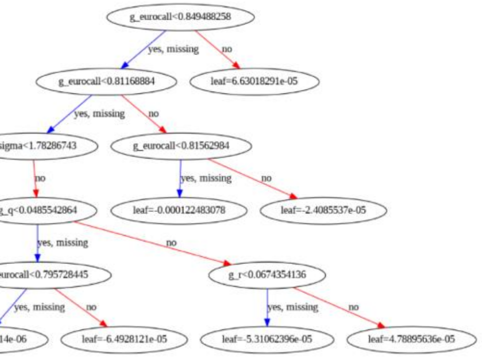
ML Pricing - European/American Option on BS Model

Extended Analysis: BS EuroCall as feature to train AmerCall



Part of the First Tree (very large tree)

The 291 Tree (Simpler structure)



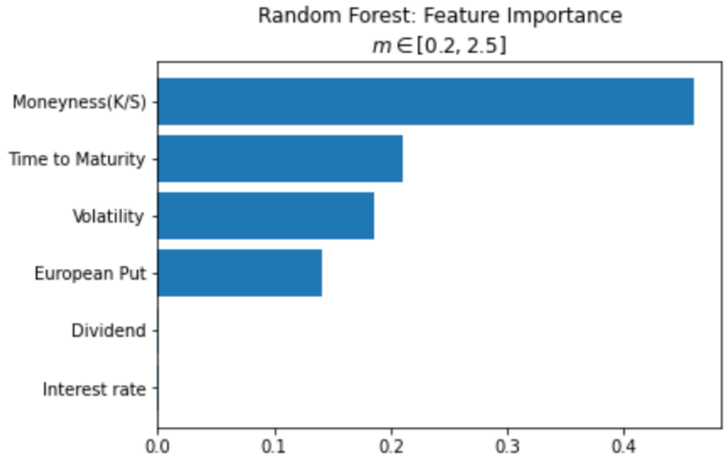
The 300 Tree (simple node tree)

- We train 800 estimators for a XGBoost Tree, Top 300 tree have complex structure and the rest of them have only simple node(This can happen when the tree has already learned all it can from the available data and no further splitting is possible.)
- We could see XGBoost still use some other features to split the nodes.
- The prediction on one data point, would run on each tree and sum up the prediction on the 800 trees.

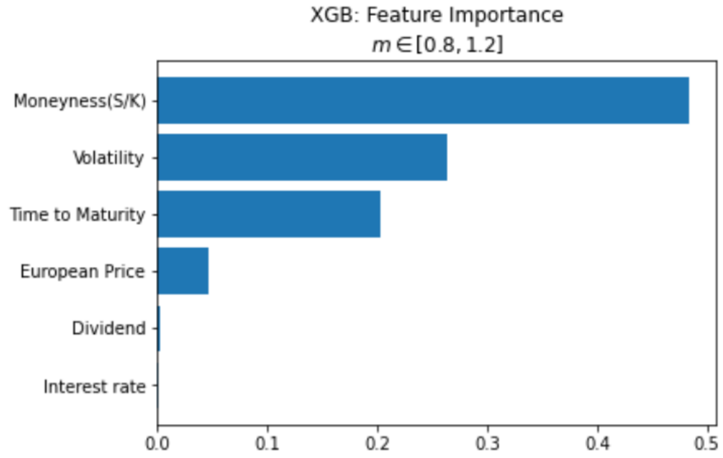
ML Pricing - European/American Option on BS Model

Extended Analysis: BS EuroPut as feature to train AmerCall

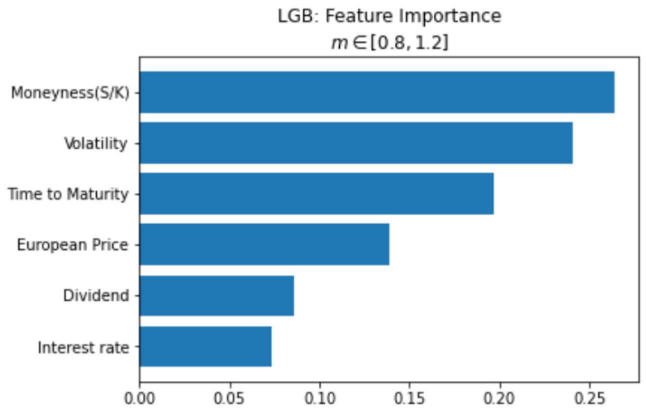
Random Forest



XGBoost



LightGBM



Motivation:

- Check if ML model could actually learn from BS price
- BS Put price should convey the same information as BS Call price

Conclusion:

- The models seem not “learn” a lot from the BS European Put Price
- The models could not yield better results than just train on *M, sigma, T, q, r*

ML Pricing - European/American Option on BS Model

Extended Analysis: BS EuroPut as feature to train AmerCall

Using BS European Put as feature:

Random Forest

RF Absolute Error (quantile)	
0.00	0.000
0.25	0.000
0.50	0.001
0.75	0.004
0.85	0.007
0.90	0.010
0.99	0.025
1.00	0.074

LightGBM

LGB Absolute Error (quantile):	
0.00	0.000
0.25	0.001
0.50	0.003
0.75	0.006
0.85	0.008
0.90	0.010
0.99	0.017
1.00	0.029

XGBoost

XGM Absolute Error (quantile):	
0.00	0.000
0.25	0.001
0.50	0.003
0.75	0.005
0.85	0.007
0.90	0.009
0.99	0.015
1.00	0.023

ANN

ANN Absolute Error (quantile):	
0.00	0.000
0.25	0.000
0.50	0.001
0.75	0.002
0.85	0.004
0.90	0.005
0.95	0.006
0.99	0.011
1.00	0.063

Using BS European Call as feature:

RF Absolute Error (quantile):	
0.00	0.000
0.25	0.000
0.50	0.000
0.75	0.000
0.85	0.001
0.90	0.001
0.99	0.005
1.00	0.032

LGB Absolute Error (quantile):	
0.00	0.000
0.25	0.000
0.50	0.000
0.75	0.001
0.85	0.002
0.90	0.002
0.99	0.005
1.00	0.045

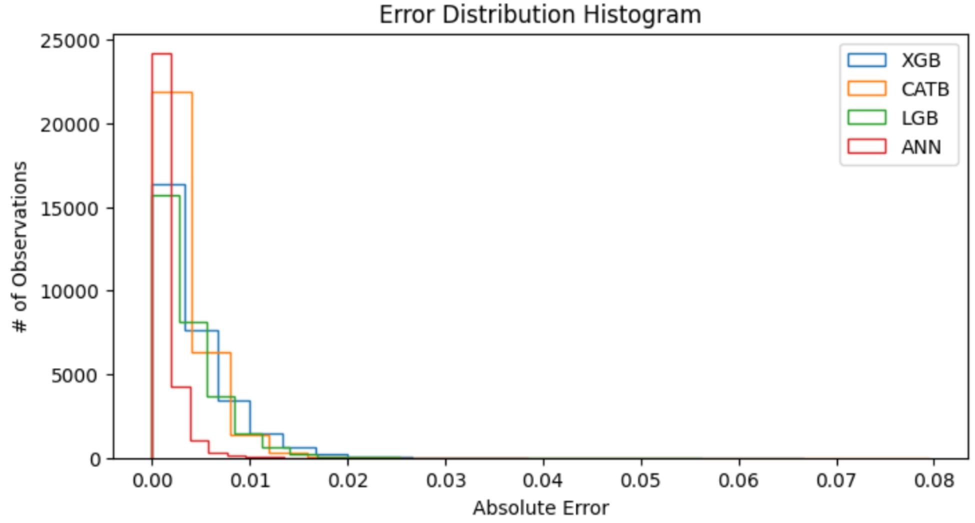
XGM Absolute Error (quantile):	
0.00	0.000
0.25	0.000
0.50	0.000
0.75	0.000
0.85	0.001
0.90	0.001
0.99	0.003
1.00	0.026

ANN Absolute Error (quantile):	
0.00	0.000
0.25	0.000
0.50	0.000
0.75	0.001
0.85	0.002
0.90	0.002
0.99	0.005
1.00	0.026

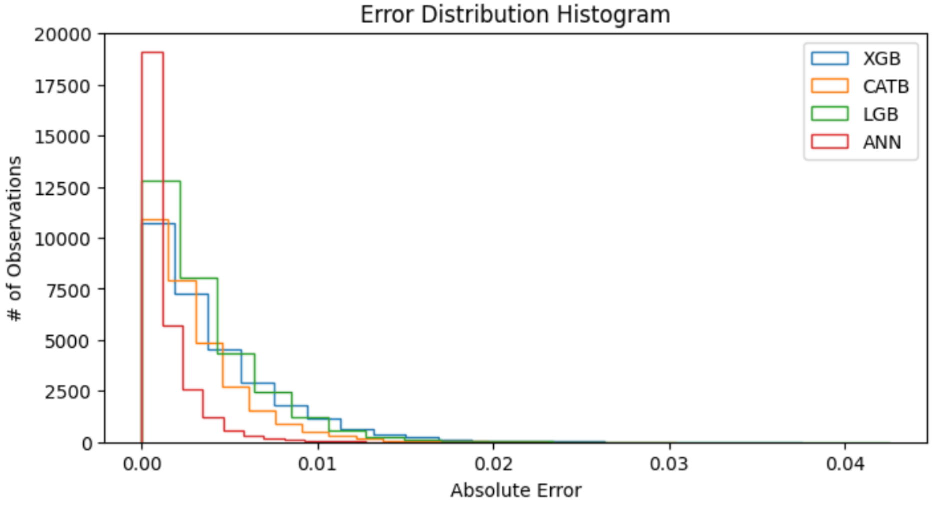
ML Pricing - European/American Option on the Heston model

Model Performance

European:



American:



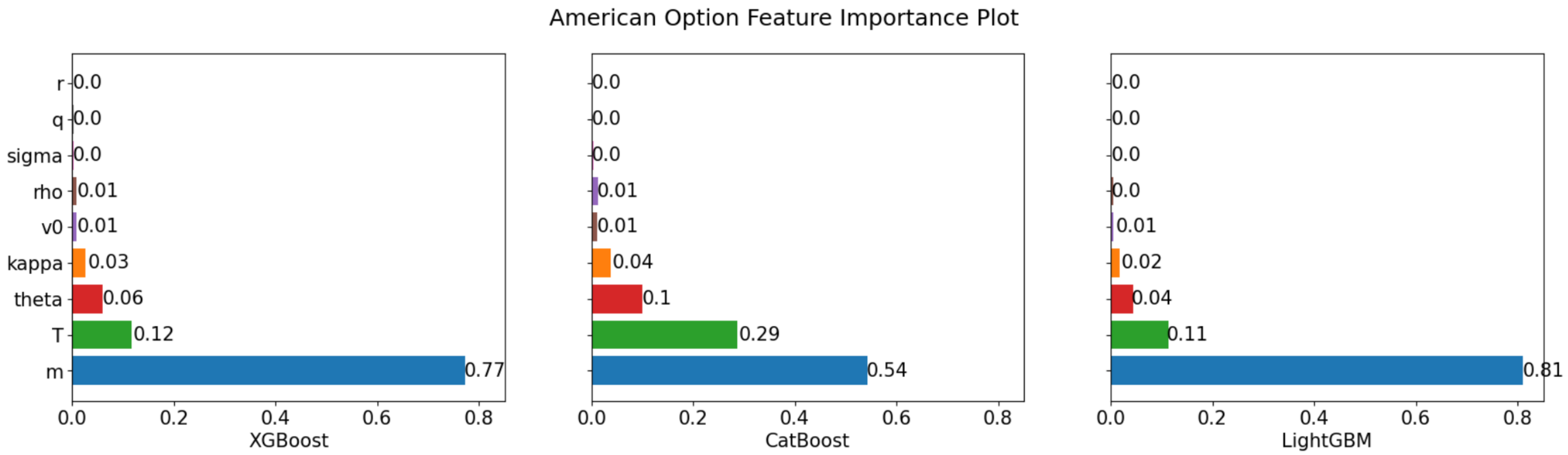
Model Performance on Test Set:

	European Call Option		American Call Option	
	RMSE	**MAE**	**RMSE**	**MAE**
XGBoost	0.0063	0.0041	0.0054	0.0039
LightGBM	0.00546	0.0036	0.0047	0.0035
CatBoost	0.00476	0.0031	0.0041	0.0029
ANN	0.0031	0.0012	0.0020	0.0013

- **Model Performance:** ANN> CatBoost ≈ LightGBM >XGBoost
- All models perform similarly for European and American Options.
- **The characteristics of ANN model make it perform better for our case study here.**

ML Pricing - European/American Option on the Heston model

Feature Importance



Conclusion:

- While XGBoost perform the best when predicting BS price among tree models, it became the worst when more features are included. (Heston model have 9 parameters)
- Still, three models share the same feature importance pattern as Moneyness being the most important one.

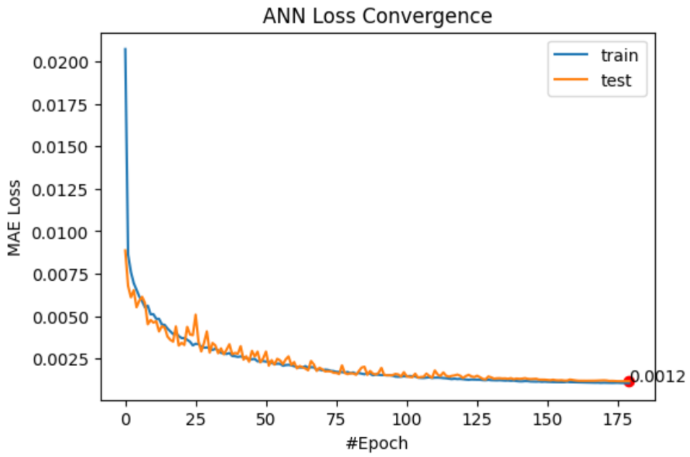
ML Pricing - European/American Option on the Heston model

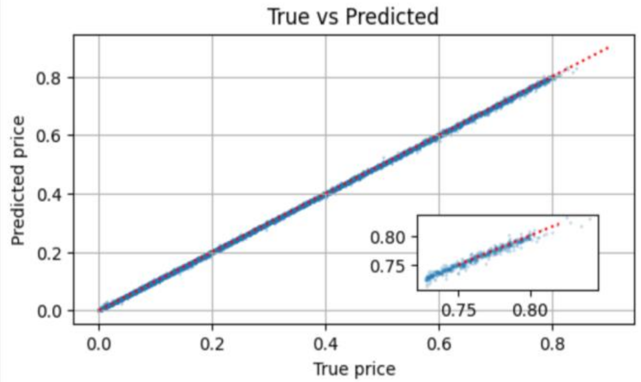
Best Model: ANN Model

Neural Network Structure:

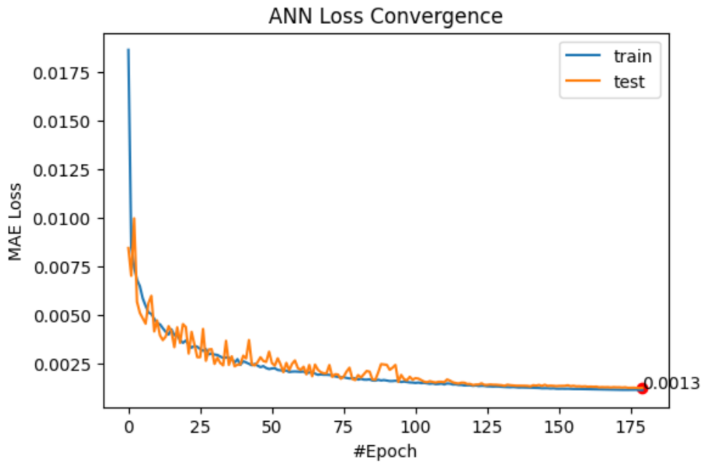
N	100,000
Batch Size	128
Initial Learning Rate	0.001
ELRA	0.98
Hidden Layers	6
Nodes per Layer	64
Epoch Size	150
Objective	MAE

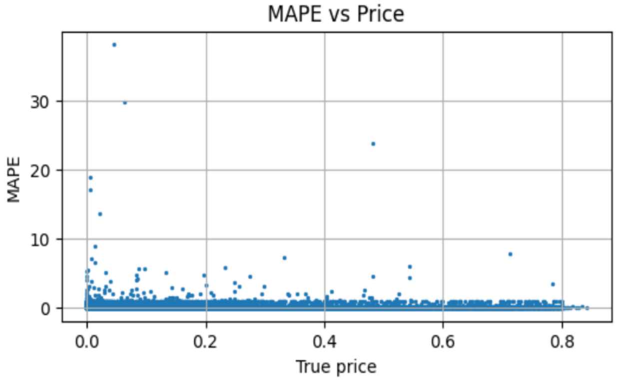
European:





American:



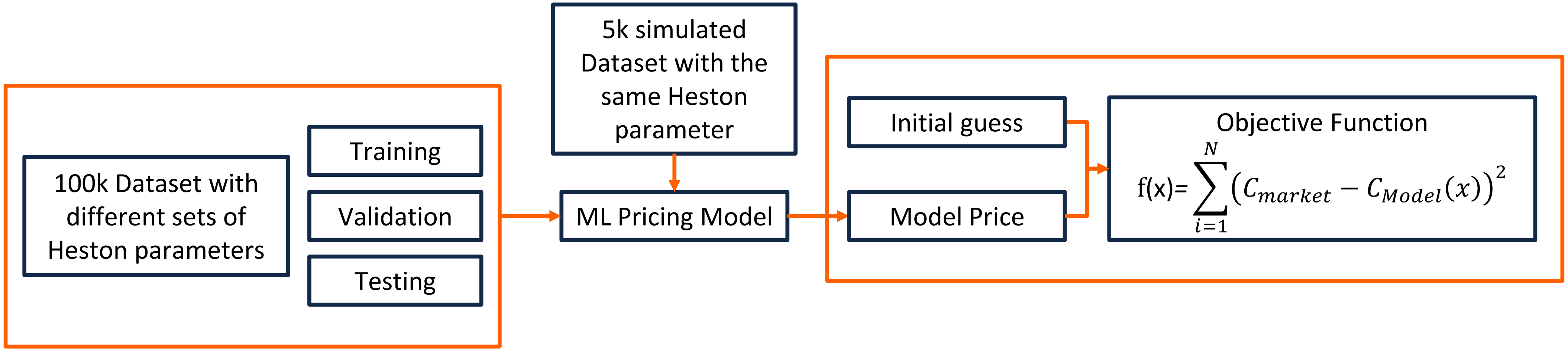


Conclusion:

- With 6 hidden layer the Mean Absolute Error for European/American Call option converge to 0.0012/0.0013.
- Loss could not further reduce with 6 hidden layers even when epoch size increased.

Parameter Calibration

Procedure



1. **Train ML pricing model**
   Train the ML model using 100k option data generated by the Fourier Transform/PDE method with the Heston model.
2. **Define the objective function**
   Market price (C_{market}): 5k option data simulated by Fourier Transform /PDE method.
   Model price ($C_{Model}(x)$): option price generated using the trained ML model with corresponding market parameter combinations and the current Heston parameters (x) as inputs.
   $\boldsymbol{x}$: a vector of Heston model parameters.
3. **Optimization**
   Use Differential Evolution (DE) optimization to get the Heston parameters.

Parameter Calibration
Heston Model

The Heston Model is based on the following stochastic differential equations:

$$dS(t) = \mu S(t)dt + \sqrt{v(t)}S(t)dW_1(t)$$
$$dv(t) = \kappa(\theta - v(t))dt + \sigma * \sqrt{v(t)}dW_2(t)$$

where:

- $S(t)$ is the underlying asset price at time t
- $v(t)$ is the stochastic volatility process at time t
- μ is the drift of the underlying asset price
- κ is the rate at which the stochastic volatility reverts to its long-term mean, θ
- θ is the long-term mean of the stochastic volatility
- σ is the volatility of the volatility
- $W_1(t)$ and $W_2(t)$ are two Wiener processes with correlation ρ, which represent the randomness in the asset price and volatility, respectively

$$\langle dW_1, dW_2 \rangle = \rho dt$$

Parameter Calibration

Heston Model

Restrictions of parameters

- $\boldsymbol{\kappa > 0}$
  This parameter represents the speed of mean reversion of the variance process, and must be positive to ensure that the variance process is mean-reverting.
- $\boldsymbol{-1 \leq \rho \leq 1}$
  This parameter represents the correlation between the Brownian motion terms in the asset price and variance processes, and must be between 0 and 1 to ensure that the processes are correlated in a meaningful way.
- $\boldsymbol{v_t > 0}$
  The variance process must be positive at all times to ensure that the asset price and its volatility are well-defined.
- $\boldsymbol{\Theta > 0}$
  This parameter represents the long-term mean of the variance process, and must be positive to ensure that the variance process does not become negative.
- $\boldsymbol{\sigma^2 < 2\kappa\theta}$
  This condition, known as the **Feller condition**, ensures that the variance process does not become negative with positive probability, which would violate the assumption of a positive variance process.

Parameter Calibration

European Option - Result

XGBoost

Group1

	v0	theta	kappa	sigma	rho	Feller
Initial	0.10000	0.05000	0.50000	0.20000	-0.70000	-0.01000
Actual	0.04000	0.04000	1.50000	0.30000	-0.50000	-0.03000
Calibrated	0.01906	0.14148	0.16222	0.20962	-0.80184	-0.00196

Group2

	v0	theta	kappa	sigma	rho	Feller
Initial	0.10000	0.05000	0.50000	0.20000	-0.70000	-0.01000
Actual	0.01000	0.02000	1.00000	0.10000	0.20000	-0.03000
Calibrated	0.01899	0.09595	0.14998	0.15905	-0.80042	-0.00349

Group3

	v0	theta	kappa	sigma	rho	Feller
Initial	0.10000	0.05000	0.50000	0.20000	-0.70000	-0.01000
Actual	0.06000	0.06000	1.50000	0.30000	-0.60000	-0.09000
Calibrated	0.01894	0.20638	0.15712	0.24504	-0.81672	-0.00481

LightGBM

	v0	theta	kappa	sigma	rho	Feller
Initial	0.10000	0.0500	0.50000	0.20000	-0.7000	-0.01000
Actual	0.04000	0.0400	1.50000	0.30000	-0.5000	-0.03000
Calibrated	0.03895	0.0723	0.07214	0.09883	0.6106	-0.00066

	v0	theta	kappa	sigma	rho	Feller
Initial	0.10000	0.05000	0.50000	0.20000	-0.70000	-0.01000
Actual	0.01000	0.02000	1.00000	0.10000	0.20000	-0.03000
Calibrated	0.01923	0.07328	0.07383	0.10181	0.90556	-0.00046

	v0	theta	kappa	sigma	rho	Feller
Initial	0.10000	0.05000	0.50000	0.20000	-0.70000	-0.01000
Actual	0.06000	0.06000	1.50000	0.30000	-0.60000	-0.09000
Calibrated	0.06562	0.09541	0.06304	0.08623	-0.82999	-0.00459

Parameter Calibration

European Option - Result

ANN

Group1

	v0	theta	kappa	sigma	rho	Feller
Initial	0.02000	0.0200	1.30000	0.1000	-0.30000	-0.04200
True	0.04000	0.0400	1.50000	0.3000	-0.50000	-0.03000
Calibrated	0.00809	0.0602	0.82162	0.3133	-0.99599	-0.00077

Group2

	v0	theta	kappa	sigma	rho	Feller
Initial	0.02000	0.01000	0.80000	0.20000	0.10000	0.02400
True	0.01000	0.02000	1.00000	0.10000	0.20000	-0.03000
Calibrated	0.00001	0.04723	0.58655	0.23535	-0.99965	-0.00001

Group3

	v0	theta	kappa	sigma	rho	Feller
Initial	0.04000	0.05000	1.30000	0.20000	-0.70000	-0.0900
True	0.06000	0.06000	1.50000	0.30000	-0.60000	-0.0900
Calibrated	0.05632	0.06884	1.03247	0.37664	-0.88684	-0.0003

Model Comparison - MAE

	Group1	Group2	Group3
XGB	0.01515	0.01905	0.01068
LightGBM	0.01059	0.01446	0.00789
ANN	0.00540	0.00662	0.00227

- **Bounds of each Heston parameter:**
  $\boldsymbol{v_0}$(1e-15, 0.5), $\boldsymbol{\theta}$(1e-15, 2), $\boldsymbol{\kappa}$(1e-15, 2),
  $\boldsymbol{\sigma}$(1e-15, 1), $\boldsymbol{\rho}$(-1, 1)
- **Model Performance:**
  ANN > LightGBM > XGBoost
- These three models have the best results for σ and the worst results for ρ.

Parameter Calibration

American Option - Result

XGBoost

Group1

	v0	theta	kappa	sigma	rho	Feller
Initial	0.02000	0.02000	1.30000	0.10000	-0.30000	-0.04200
Actual	0.04000	0.04000	1.50000	0.30000	-0.50000	-0.03000
Calibrated	0.02362	0.02521	0.23019	0.03595	0.39984	-0.01031

Group2

	v0	theta	kappa	sigma	rho	Feller
Initial	0.02000	0.01000	0.80000	0.20000	0.10000	0.0240
Actual	0.01000	0.02000	1.00000	0.10000	0.20000	-0.0300
Calibrated	0.03705	0.02434	0.04677	0.04221	0.61837	-0.0005

Group3

	v0	theta	kappa	sigma	rho	Feller
Initial	0.04000	0.05000	1.30000	0.20000	-0.70000	-0.09000
Actual	0.06000	0.06000	1.50000	0.30000	-0.60000	-0.09000
Calibrated	0.04039	0.19635	0.23006	0.12062	-0.78153	-0.07579

LightGBM

	v0	theta	kappa	sigma	rho	Feller
Initial	0.0200	0.02000	1.30000	0.10000	-0.30000	-0.04200
Actual	0.0400	0.04000	1.50000	0.30000	-0.50000	-0.03000
Calibrated	0.0367	0.04053	0.17582	0.11436	0.77303	-0.00117

	v0	theta	kappa	sigma	rho	Feller
Initial	0.02000	0.01000	0.80000	0.20000	0.10000	0.02400
Actual	0.01000	0.02000	1.00000	0.10000	0.20000	-0.03000
Calibrated	0.00062	0.01292	0.16416	0.02056	-0.76423	-0.00382

	v0	theta	kappa	sigma	rho	Feller
Initial	0.04000	0.05000	1.30000	0.20000	-0.70000	-0.09000
Actual	0.06000	0.06000	1.50000	0.30000	-0.60000	-0.09000
Calibrated	0.04793	0.08662	0.16522	0.12873	-0.78071	-0.01205

Parameter Calibration

American Option - Result

ANN

Group1

	v0	theta	kappa	sigma	rho	Feller
Initial	0.10000	0.0500	0.5000	0.2000	0.3000	-0.01000
Actual	0.04000	0.0400	1.5000	0.3000	-0.5000	-0.03000
Calibrated	0.02768	0.2284	0.2986	0.3692	-0.4991	-0.00009

Group2

	v0	theta	kappa	sigma	rho	Feller
Initial	0.10000	0.0500	0.5000	0.2000	0.3000	-0.01000
Actual	0.01000	0.0200	1.0000	0.1000	0.2000	-0.03000
Calibrated	0.00272	0.1702	0.1857	0.2513	-0.4999	-0.00006

Group3

	v0	theta	kappa	sigma	rho	Feller
Initial	0.10000	0.0500	0.5000	0.200	0.3000	-0.01000
Actual	0.06000	0.0600	1.5000	0.300	-0.6000	-0.09000
Calibrated	0.05283	0.2717	0.3263	0.421	-0.4999	-0.00007

Model Comparison - MAE

	Group1	Group2	Group3
XGB	0.00679	0.00922	0.00669
LightGBM	0.00858	0.01218	0.00640
ANN	0.00489	0.00511	0.00210

- **Bounds of each Heston parameter:** $\boldsymbol{v_0}$(1e-15, 0.5), $\boldsymbol{\theta}$(1e-15, 2), $\boldsymbol{\kappa}$(1e-15, 2), $\boldsymbol{\sigma}$(1e-15, 1), $\boldsymbol{\rho}$(-1, 1)
- **Model Performance:** ANN > XGBoost > LightGBM
- These three models have the best results for σ and the worst results for ρ (the same as European).

Parameter Calibration

Result

		v_0	θ	κ	σ	ρ	MAE
XGBoost	**European**	70.2%	292.5%	87.9%	35.8%	198.9%	0.015
	American	114.7%	95.3%	88.2%	68.5%	139.8%	0.008
LightGBM	**European**	34.8%	135.4%	94.5%	46.7%	204.4%	0.011
	American	40.7%	27.0%	86.9%	66.1%	255.6%	0.009
ANN	**European**	61.9%	67.1%	39.2%	55.1%	248.9%	0.005
	American	38.5%	524.9%	79.9%	71.6%	122.3%	0.004

- The percentage error of the Heston parameters in the table is calculated by $\left|\frac{P_{Calibrated}-P_{True}}{P_{True}}\right|$, where P represents the Heston parameters, i.e. $P_{Calibrated} = v_{0_{Calibrated}}$.
- The mean absolute error (MAE) is calculated by $|C_{Market} - C_{Model}|$, where C represents the option price.
- **Performance:** ANN > XGBoost ≈ LightGBM; American > European
- The results of the estimation of these Heston parameters are not good enough.
- **Some possible causes of the poor performance:**
  1. The ML pricers may not good enough. If the pricers can be more accurate, the percentage error could be close to 0.
  2. The optimizer may be trapped in a local solution.
  3. There may exist several parameter combinations leading to the same result.

Parameter Calibration

Dig into possible causes of the poor performance

	diff_calibrated	diff_true
XGB-Group1	0.00679	0.01167
XGB-Group2	0.00922	0.01520
XGB-Group3	0.00669	0.00999
LGB-Group1	0.00858	0.01238
LGB-Group2	0.01218	0.01645
LGB-Group3	0.00640	0.01042

- **Definition:**

$$diff\_calibrated = \frac{1}{N}\sum |C_{market} - C_{model}(x)|$$

$$diff\_true = \frac{1}{N}\sum |C_{market} - C_{true}|$$

C_{market}: option data simulated by Fourier Transform/PDE method
$C_{model}(x)$: option price generated using the trained ML model with corresponding market parameter combinations and the calibrated Heston parameters (x) as inputs
C_{true}: option price generated using the trained ML model with the true Heston parameters used to simulated the C_{market}
x: a vector of calibrated Heston model parameters.

- **Result:** $diff\_calibrated < diff\_true$
- **Conclusion:**
  1. The optimizer did work.
  2. The result comes from our ML pricers are not accurate enough, which gives our optimizer the possibility to get closer to our simulated market price, C_{market}.
  3. More accurate ML pricers will eliminate the $diff\_true$, which makes our calibration more accurate.

Parameter Calibration

Time Efficiency Comparison

	Traditional Method (Fourier Transform)	XGBoost	LightGBM	ANN
CPU Time	15.3s	4.98s	5.11s	40.4ms

- Running the Tradition pricer 5 times would result in CPU times as 15.3sm which is about ***3x*** comparing with XGBoost and LightGBM, and ***383x*** comparing with ANN.
- The calibration time using tree models (XGBoost/LightGBM)/ANN is therefore about 48 minutes/33 seconds, that is ***87x***.
- Calibrating parameters using ML pricers greatly promotes the efficiency.

Findings

- ANN model is sensitive to the size of the training set while tree models are less sensitive.
- All models‘ performances show little difference for European/American Option on simulated BS/Heston datapoints.
- The ANN model displays superior predictive power for option pricing when compared to other widely-used tree models. As such, we believe that further investigation into methods of increasing its accuracy is warranted.
- To assess the feasibility of using NN as a pricer to aid with model calibration, we must continue to improve upon the model's capabilities and test its performance using real-world market data.

Future Work

ANN Calibration without traditional optimizer

Reference:

[HTML] A neural network-based framework for financial model calibration
S Liu, A Borovykh, LA Grzelak… - Journal of Mathematics in …, 2019 - Springer
A data-driven approach called CaNN (Calibration Neural Network) is proposed to calibrate financial asset price models using an Artificial Neural Network (ANN). Determining optimal values of the model parameters is formulated as training hidden neurons within a machine learning framework, based on available financial option prices. The framework consists of two parts: a forward pass in which we train the weights of the ANN off-line, valuing options under many different asset model parameter settings; and a backward pass, in which we …
☆ Save 99 Cite Cited by 81 Related articles All 22 versions

[HTML] Deep calibration of financial models: turning theory into practice
P Büchel, M Kratochwil, M Nagl, D Rösch - Review of Derivatives …, 2022 - Springer
The calibration of financial models is laborious, time-consuming and expensive, and needs to be performed frequently by financial institutions. Recently, the application of artificial neural networks (ANNs) for model calibration has gained interest. This paper provides the first comprehensive empirical study on the application of ANNs for calibration based on observed market data. We benchmark the performance of the ANN approach against a real-life calibration framework that is in action at a large financial institution. The ANN based …
☆ Save 99 Cite Cited by 3 Related articles All 10 versions

Method Introduction

The first paper introduced Calibration Neural Network(3 phases):

- First Phase: Mapping from 9 parameters to price.(train the model to best fit the result) ➡ Forward Pass
- Second Phase: Infer from the model.(apply the model to test-set to generate prediction)
- Third Phase: “Inverse” the trained ANN model to try to train the input layer?(details on next slide) ➡ Backward Pass

Future Work

ANN Calibration without traditional optimizer

From the author

During the calibration phase (or the backward pass), the original input layer of the ANN is transformed into a learnable layer, while all hidden layers remain unchanged. These layers are the ANN layers obtained from the forward pass with the already trained weights. By providing the output data, here consisting of market-observed option prices and implied volatilities, and changing to an objective function for model calibration, the ANN can be used to find the input values that match the given output. **The task is thus to solve the inverse problem by learning a certain set of input values, here the model parameters Θ, either for the Heston or Bates model.**

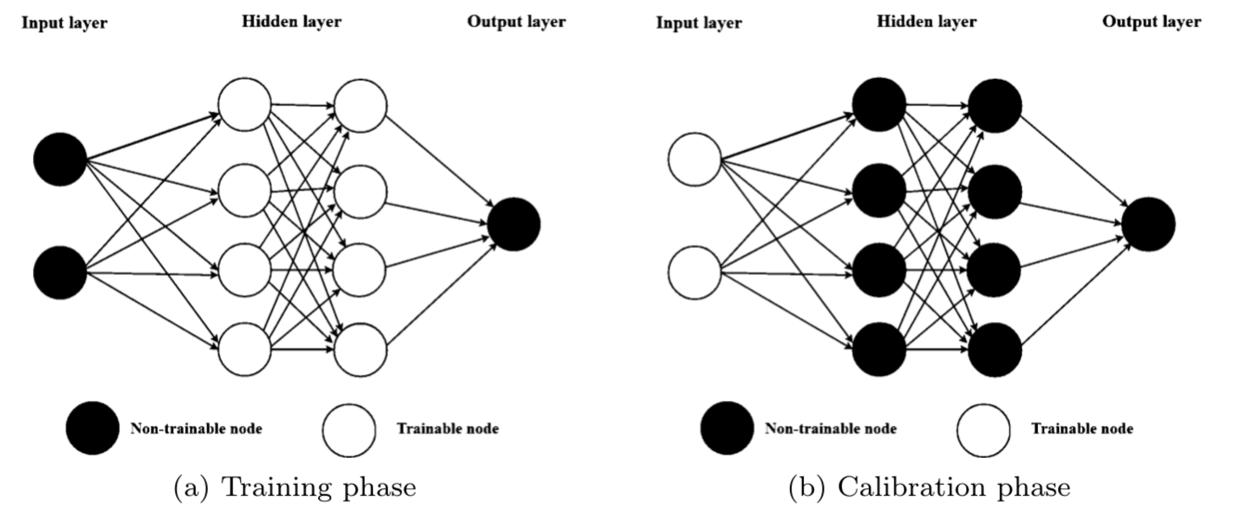


Figure 2 The different phases of the CaNN

Potential future work

- How to implement this in reality: inverse a NN model?
- The first layer in our case has shape (64, 9) which should be frozen as the author described.