A detailed abstract illustration of a futuristic city. It features a variety of skyscrapers with unique, rounded, and organic shapes. The buildings are primarily in shades of blue, teal, and light orange. A complex network of glowing yellow and white lines, resembling circuit boards or neural networks, weaves through the city, connecting different parts of the architecture. In the upper right, a large, dark rectangular structure with a grid pattern, possibly representing a server or a large screen, is visible. The sky above is filled with small, colorful dots representing stars or data points. A small, stylized figure of a person stands on a path in the lower center-left. A digital interface element in the upper left corner displays the number "10,598,000" in a yellow box with a small upward arrow icon.

# Machine Learning for Options Pricing

Predicting SPY ETF maximum prices and estimating call option fair values using advanced machine learning techniques.

## Group A

- **Michael Brick**
- **Meng Li**
- **Sharath Mohan Kumar**

# Project Overview

What is a call option?

A derivative contract that gives the owner the right to buy an underlying security (a stock, bond, or Exchange Traded Fund [ETF]) at a certain specified "strike" price up until a certain expiration date.

Acts as a form of leverage, increasing and decreasing in value much faster (proportionately) than the underlying security itself.

Valuing options contracts precisely is a notoriously difficult problem. The most famous currently prevailing model is called "Black-Scholes."



# Project Overview

Our project attempts to approach the problem using machine learning concepts and methods.

Focuses on predicting most likely maximum price to be reached not at expiration, but at any point between purchase and expiration.

Better reflects **actual trading behavior** and market dynamics.



# Black-Scholes Baseline

## Traditional Model

We implemented the Black-Scholes formula as a baseline for comparison:

$$C = S \cdot e^{-qT} \cdot N(d_1) - K \cdot e^{-rT} \cdot N(d_2)$$

Where annualized volatility, prevailing interest rates, dividend yields, time to expiration, and normal probability distributions drive the calculation.

# Data Sources

## Options Data

Sourced from OptionMetrics via Rutgers Libraries.

## SPY ETF Pricing, Interest Rates & Dividends

Obtained from S&P Capital IQ.

# Features Used in the Model

## Base Features

- SPY Price
- Days to expiration
- Dividend Yield
- Risk-free rate
- Standard deviation (volatility)
- SD of SD

## Lookback Features (1-42 days)

- Max - Back
- Min - Back
- Delta Back

# Data Processing Pipeline



## Filtering

Strike prices within 1% of SPY price, expiration within 6 weeks, call options only.



## Feature Selection

132 features including base metrics, 42-day lookback windows for max/min/delta values.



## Cleaning

Remove missing values, handle NaN/infinite values for stability.



## Dataset Overview

**Final Dataset: 145,580 rows and 221 columns**

# Machine Learning Methods



## Multi-Layer Perceptron

Fully connected neural network with 256-unit layers and dropout regularization.



## LSTM Networks

Long Short-Term Memory architecture for capturing temporal dependencies in price movements.



## Support Vector Regression

RBF kernel-based regression with standardized features for robust predictions.



## Gradient Boosting

LightGBM with grid search optimization for maximum predictive accuracy.

# MLP Model for Forward Max Price Prediction

## 1 Objective

Predict forward maximum price using a Multi-Layer Perceptron (MLP) neural network.

## 2 Architecture

Fully connected network with 256-unit dense layers (ReLU activation, 0.2 dropout), a 128-unit layer, and a 1-unit output layer for regression.

## 3 Training Configuration

Adam optimizer, MAE loss function, 50 epochs, and a batch size of 512.

# MLP Training Details

## Data Preparation

- Train-Test Split (80% training, 20% testing with random seed 42)
- Feature Standardization using StandardScaler

## Training Configuration

- Optimizer Adam (learning rate 0.001)
- Loss function MAE
- Validation split 20%
- Epochs 50
- Batch size 512
- Early stopping with patience 5

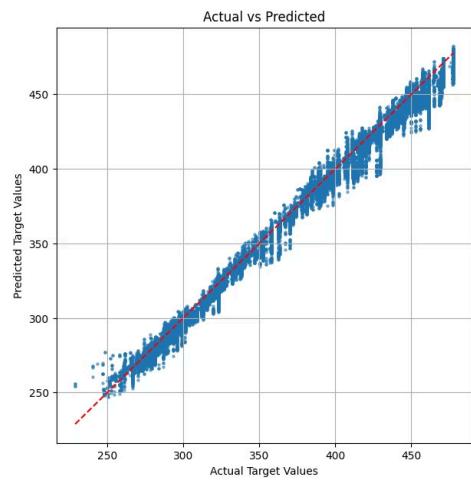
## Evaluation Metrics

- MAE (Mean Absolute Error)
- RMSE (Root Mean Squared Error)

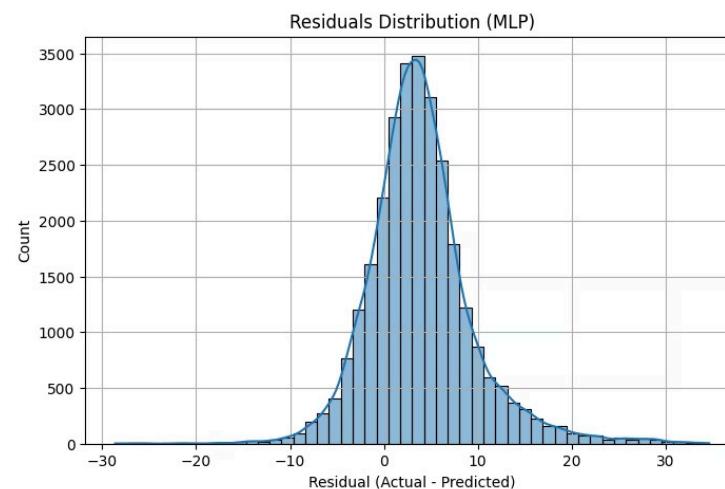
# MLP Results

Test MAE: 5.1479, RMSE: 6.9084

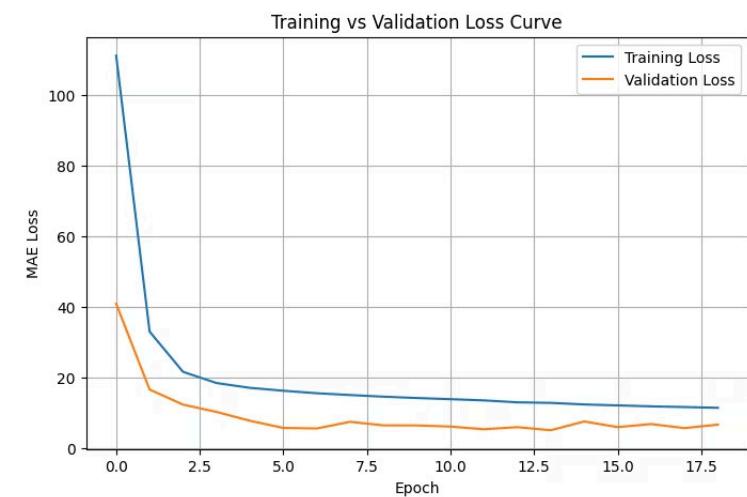
Actual vs. Predicted



Residuals Distribution



Training History



# LSTM Model for Forward Max Price Prediction

## Objective

Use LSTM neural network to predict forward maximum price.

## Architecture

The model architecture includes:

- LSTM layer: 64 units
- Dropout: 0.3
- Dense layers: 64 units (ReLU), 32 units (ReLU)
- Output layer: 1 unit (regression)

## Training Configuration

The model was trained with:

- Optimizer: Adam
- Loss function: MAE
- Epochs: 20
- Batch size: 512

# LSTM Training Details

## Data Preparation

- Train-Test Split (80% training, 20% testing with random seed 42)
- Feature Standardization using StandardScaler
- Reshape for LSTM (3D format with timesteps = 1)

## Training Configuration

- Optimizer Adam
- Loss function MAE
- Validation split 20%
- Epochs 20
- Batch size 512

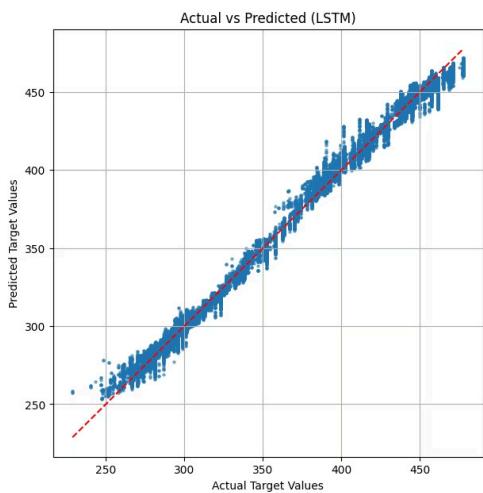
## Evaluation Metrics

- MAE (Mean Absolute Error)
- RMSE (Root Mean Squared Error)

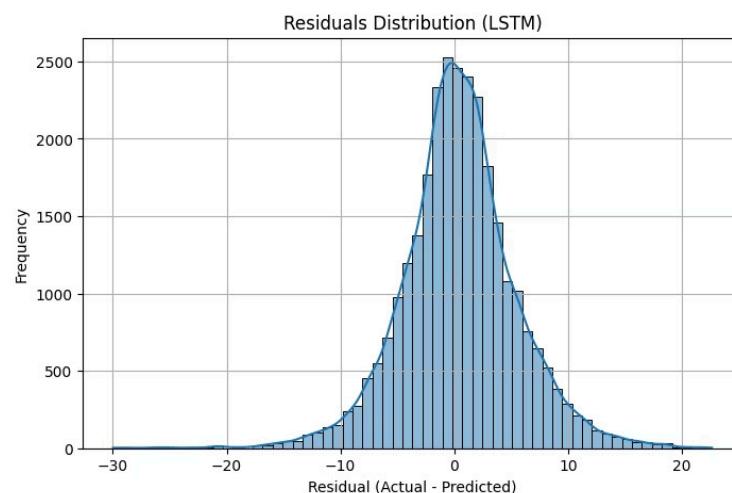
# LSTM Results

Test MAE: 3.7786, RMSE: 5.0878

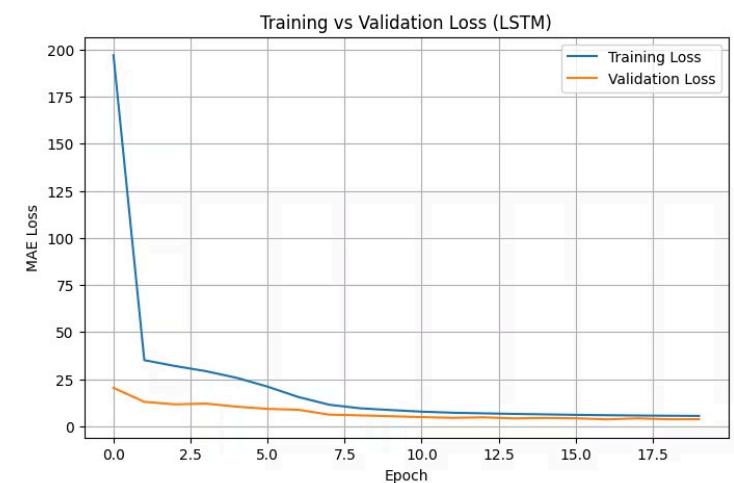
## Actual vs. Predicted



## Residuals Distribution



## Training History



# SVR Model for Forward Max Price Prediction

## 1 Objective

Utilize Support Vector Regression (SVR) to predict the forward maximum price.

## 2 Model Choice

- A fundamentally different approach from neural networks – finds a hyperplane to fit the data
- RBF kernel captures nonlinear relationships well, potentially improving prediction accuracy

## 3 Training Configuration

- Batch Prediction:  
Predictions on the test set in batches (size = 1000).

# SVR Training Details

## Data Preparation

- Train-Test Split (80% training, 20% testing)
- Feature Standardization using StandardScaler for both features and target

## Model Configuration

- Kernel RBF
- C: 1.0
- Epsilon: 0.1
- Gamma: scale
- Cache size: 2000

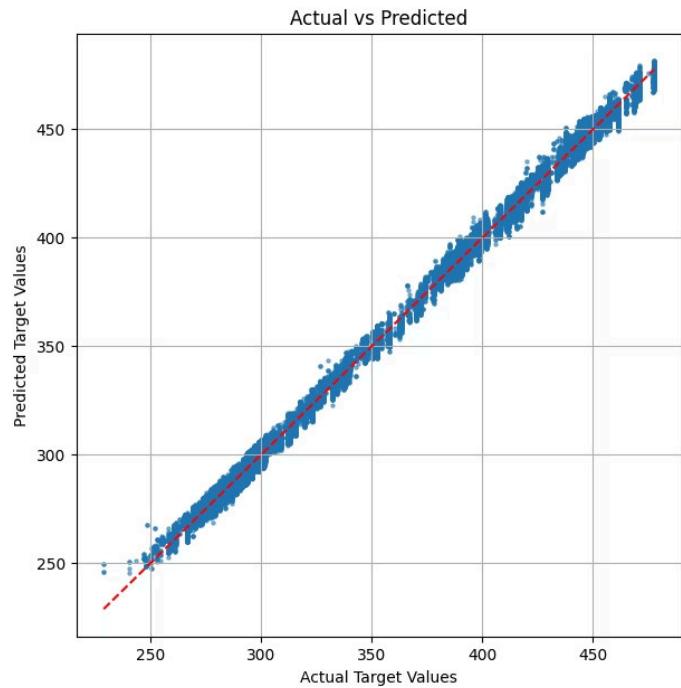
## Evaluation Metrics

- MAE (Mean Absolute Error)
- RMSE (Root Mean Squared Error)

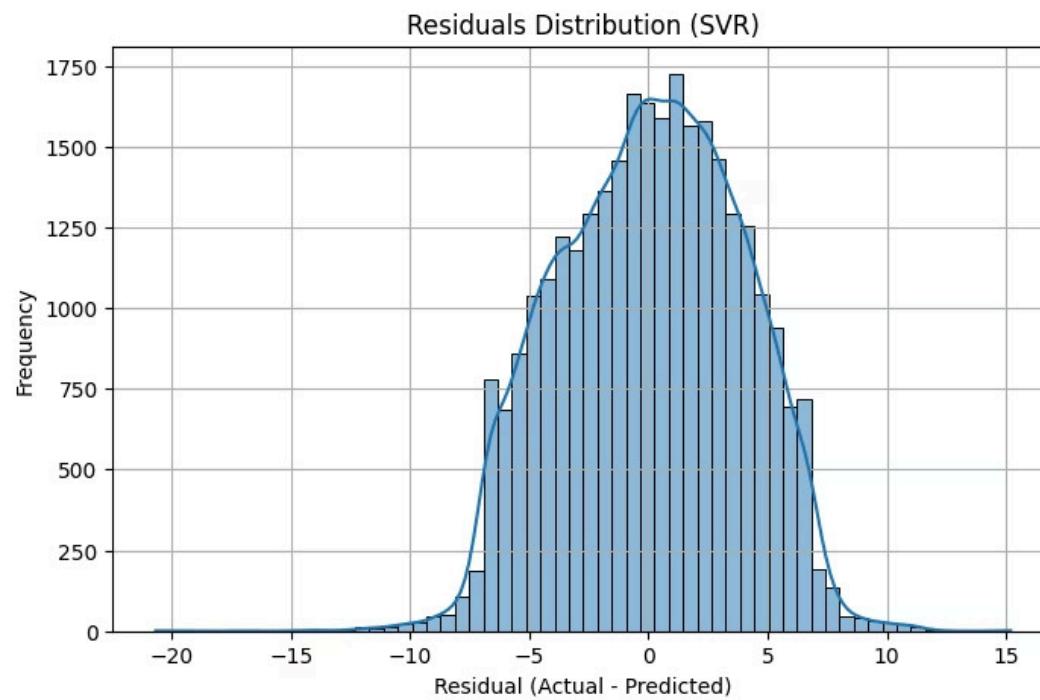
# SVR Results

Test MAE: 3.1168, RMSE: 3.7726

Actual vs. Predicted Values



Residuals Distribution



# LightGBM Model for Forward Max Price Prediction

## Objective

Use LightGBM gradient boosting to predict forward maximum price.

## Model Choice

- Efficient with large datasets, captures non-linear relationships and feature interactions automatically
- Gradient boosting consistently outperforms traditional methods for complex financial data

## Training Configuration

StandardScaler for feature normalization, early stopping with validation set.

# LightGBM Training Details

## Data Preparation

- Train-Test Split (80% training, 20% testing)
- Feature Standardization using StandardScaler

## Model Configuration

- Gradient Boosting
- Number of estimators: 100
- Max depth: 7
- Learning rate: 0.1
- Subsample: 0.8
- Early stopping with validation set

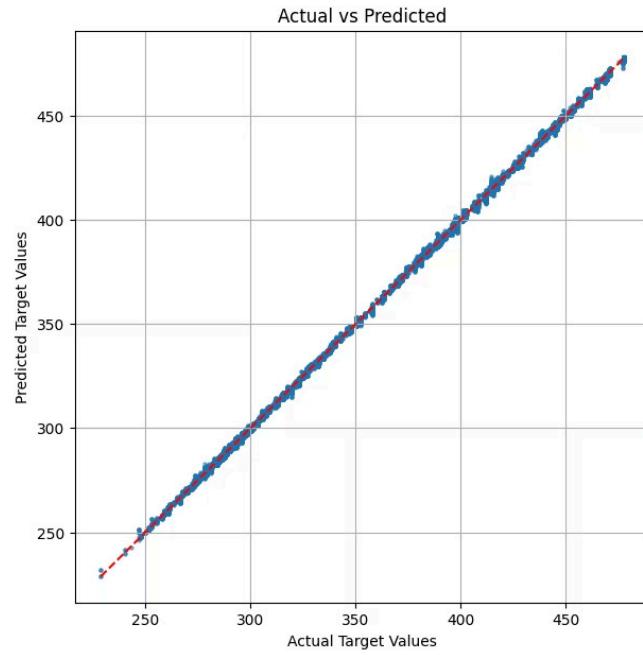
## Evaluation Metrics

- MAE (Mean Absolute Error)
- RMSE (Root Mean Squared Error)

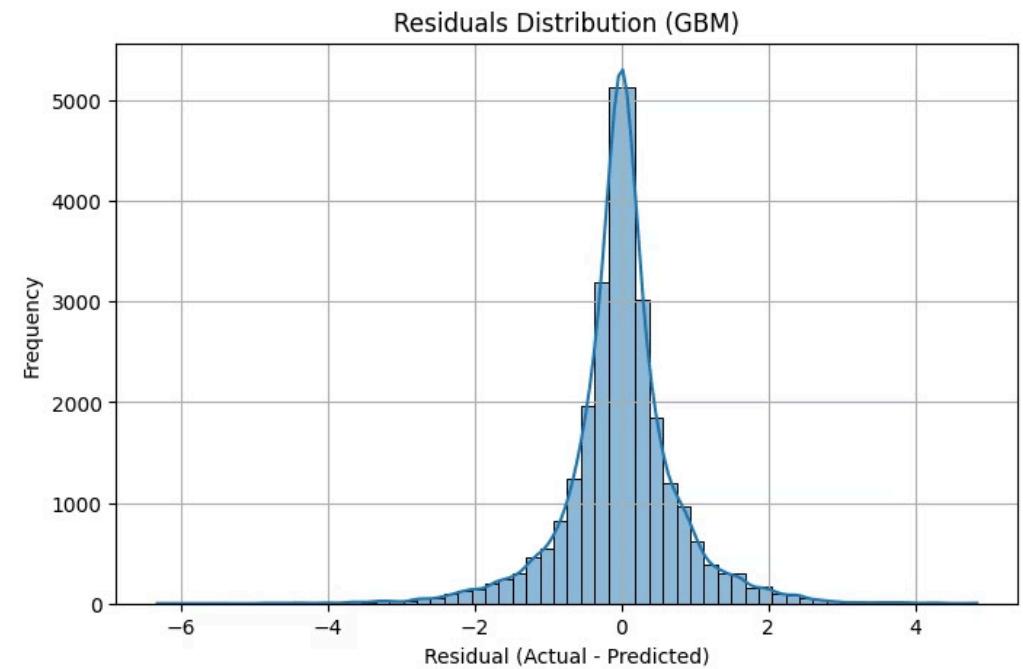
# LightGBM Results

Test MAE: 0.5021, RMSE: 0.7610

Actual vs. Predicted Values



Residuals Distribution



# Model Performance Comparison

Model	MAE	RMSE
MLP	5.1479	6.9084
LSTM	3.7786	5.0878
SVR	3.1168	3.7726
LightGBM	0.5021	0.7610

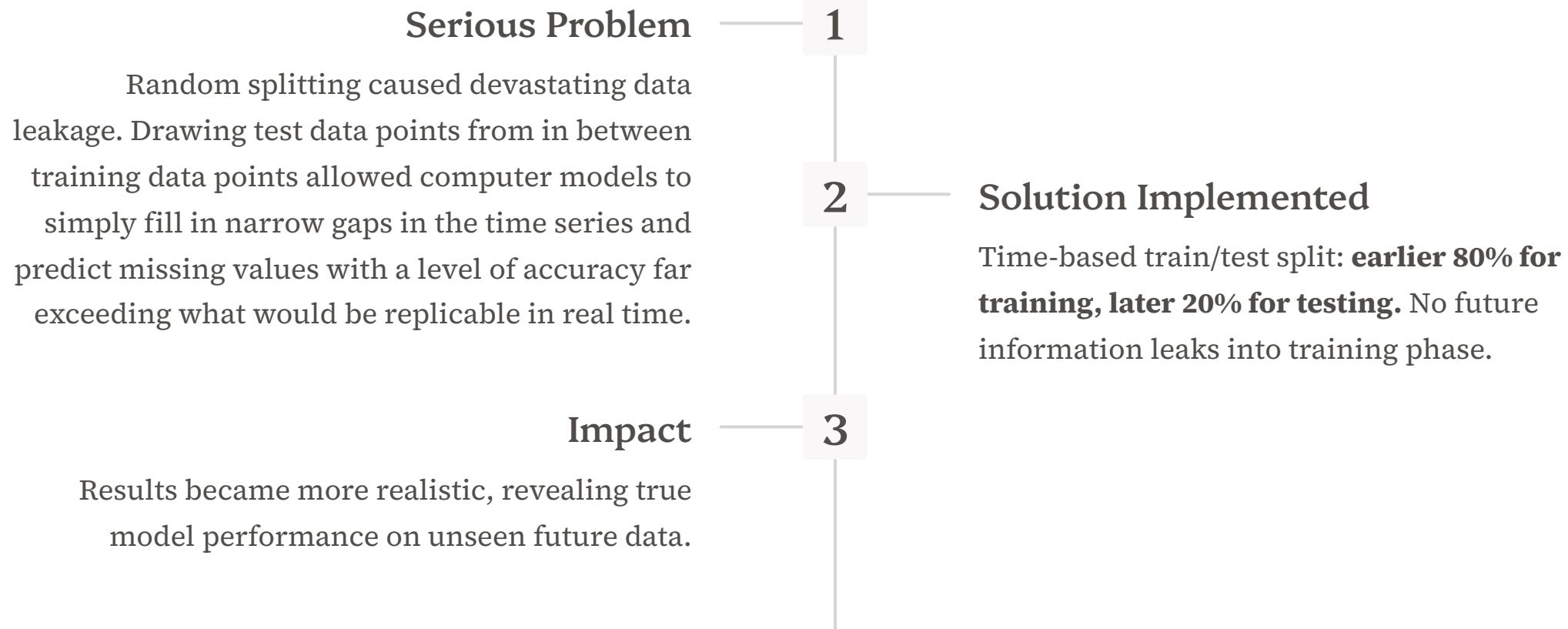
These updated test results clearly show that **LightGBM** dramatically outperforms all other models, achieving significantly better MAE and RMSE scores.

# Data Overview

Dates	target	BS_call	pred_MLP	pred_LSTM	pred_SVR	pred_GBM	
May-25-2018	<b>278.92</b>	4.363668	283.03	285.64	282.34	278.89	<b>Train Set</b>
May-25-2018	<b>278.92</b>	3.461110	NaN	NaN	NaN	NaN	<b>Test Set</b>
May-25-2018	<b>278.92</b>	3.677894	NaN	NaN	NaN	NaN	<b>Test Set</b>
May-25-2018	<b>278.92</b>	3.903769	NaN	NaN	NaN	NaN	<b>Test Set</b>
May-25-2018	<b>278.92</b>	4.138800	282.30	285.12	281.69	278.85	<b>Train Set</b>

Upon reviewing the data, we discovered that **observations from the same date** were split across both training and testing sets, potentially causing **data leakage of the target variable**.

# Critical Discovery: Data Leakage



# Model Performance Using Time-Based Split

Data Split: Earlier 80% for training, later 20% for testing

Model	MAE	RMSE
MLP	23.2	27.92
LSTM	22.5	26.99
SVR	19.54	24.72
LightGBM	10.74	13.69

**LightGBM emerged as the clear winner** with Test MAE of 10.74 and RMSE of 13.69.

To maximize LightGBM's predictive power, we employed **grid search optimization** across **4 key hyperparameters** to identify the optimal configuration.

# Grid Search Optimization of LightGBM

- **Method:** Grid search with early stopping (patience = 50 rounds)
- **Validation metric:** MAE (Mean Absolute Error)

## Search Space

- num\_leaves: 55, 70 learning\_rate: 0.015, 0.025
- feature\_fraction: 0.65, 0.75 lambda\_l1: 0.05, 0.15

## Optimal Parameters

- num\_leaves: 70, learning\_rate: 0.025, feature\_fraction: 0.65, lambda\_l1: 0.05

## Validation Performance

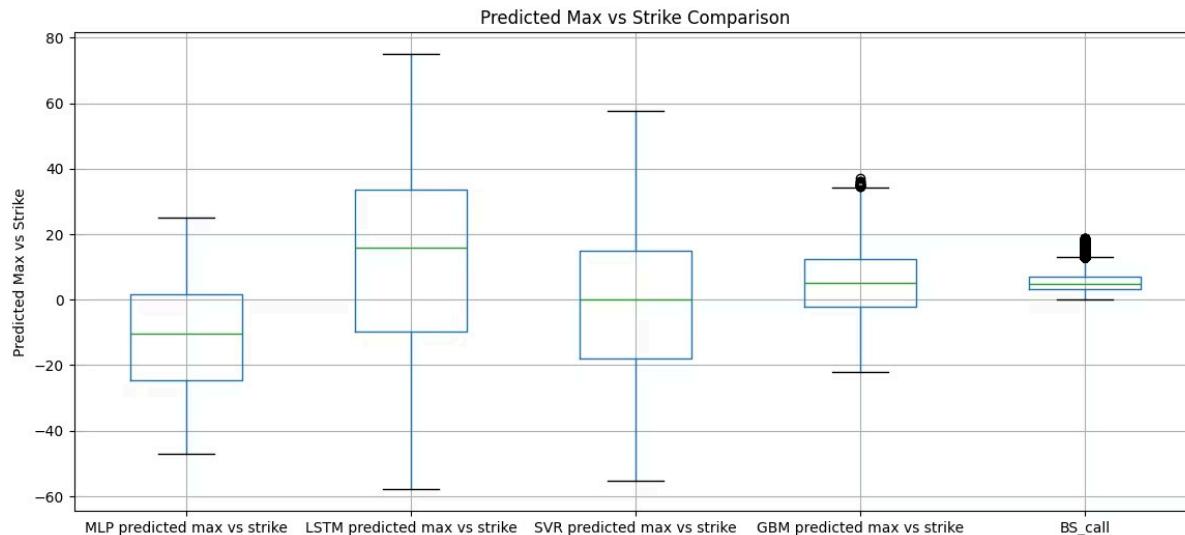
- MAE = 10.74, RMSE = 13.69

# Descriptive Statistics

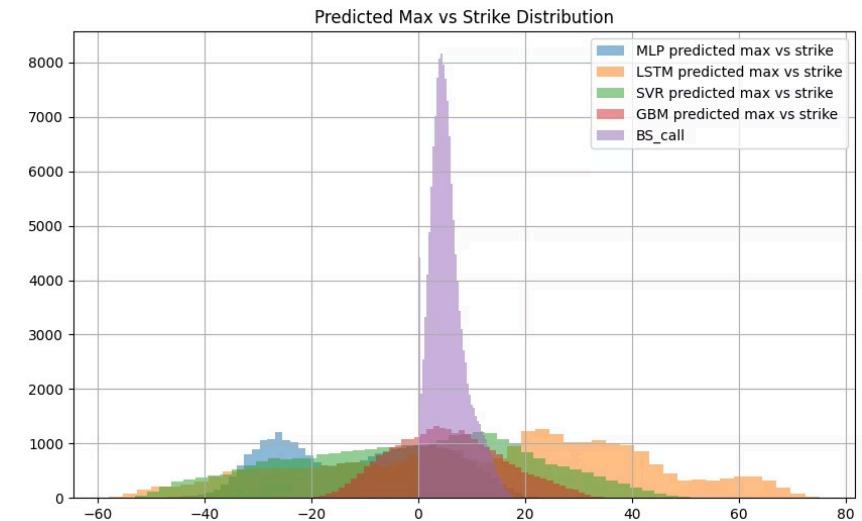
	<b>MLP</b>	<b>LSTM</b>	<b>SVR</b>	<b>GBM</b>	<b>BS_call</b>
<b>Mean</b>	-11.0267	11.9865	-1.5032	5.5434	5.4875
<b>Std</b>	14.9374	28.8947	21.9160	10.2735	3.3114
<b>Min</b>	-47.0002	-57.8783	-55.2653	-22.0517	0.0000
<b>25%</b>	-24.6152	-9.8643	-18.0943	-2.0692	3.1890
<b>50%</b>	-10.3311	15.9546	0.1962	5.0405	4.9329
<b>75%</b>	1.6365	33.5622	14.8520	12.5400	7.1686
<b>Max</b>	25.1332	75.0132	57.7500	37.0639	18.8781
<b>Mean Diff vs BS</b>	-18.2149	4.7992	-8.6915	-1.6440	0.0000
<b>Corr with BS</b>	-0.1149	0.3177	0.0976	0.3806	1.0000

# Comparison of Model Predictions vs. Strike Price

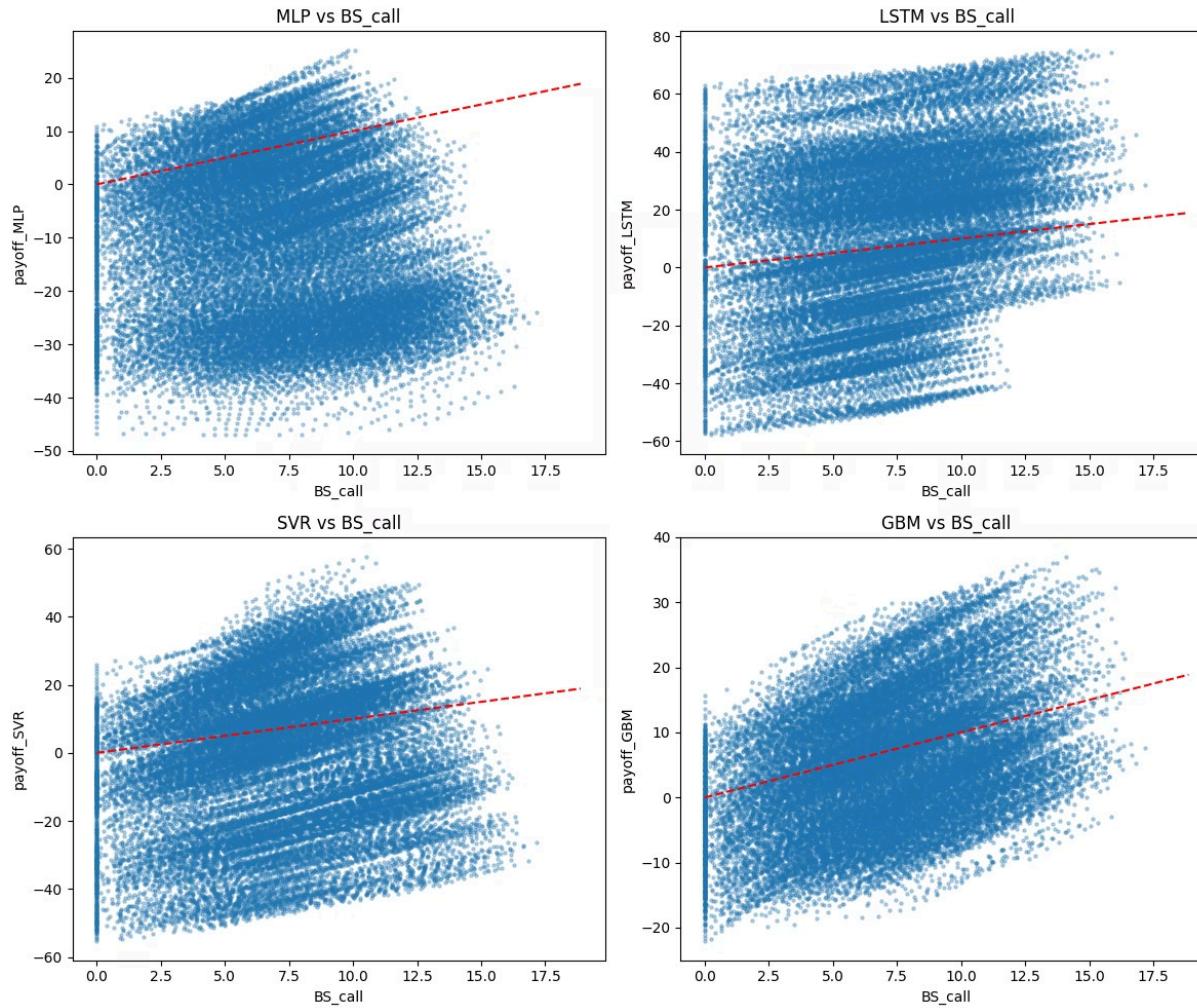
## Boxplot



## Distribution



# ML Predicted vs. Black-Scholes Call Value



# Project Summary

## Project Overview

- Predicted SPY ETF maximum prices using ML to estimate call option fair values
- Adopted a maximum likelihood framework that better reflects actual trading behavior

## Methods Used

- ML Models: MLP, LSTM, SVR, GBM
- Black-Scholes (benchmark)

## Advanced Techniques

- LSTM for time-series prediction
- Maximum likelihood framework for options pricing
- Grid search for hyperparameter tuning

# Team Contributions



## Michael Brick

Project conception, data acquisition from OptionMetrics and S&P Capital IQ, volatility metrics computation.



## Meng Li

Data cleaning and processing, GBM and SVR implementation and evaluation, grid search for hyperparameter tuning.



## Sharath Mohan Kumar

MLP and LSTM neural network implementation and evaluation, graphic projections python coding.

# Thank You for Your Attention

**Group A**

Michael Brick

Meng Li

Sharath Mohan Kumar