Contents

Electricity Price Forecasting - Presentation Talking Points

---

1. INTRODUCTION

Problem Statement: - Electricity prices are highly volatile and difficult to predict - Accurate forecasting enables better trading decisions and risk management - Goal: Predict daily price direction (Up/Down) for PJM electricity market

Dataset: - PJM (Pennsylvania-New Jersey-Maryland) electricity market data (2002-2024) - 8,020 daily samples - 27 engineered features: price, volume, gas prices, temperature, load, technical indicators - 70/15/15 train/val/test split

---

2. MODELS IMPLEMENTED

LightGBM (Gradient Boosting)

- Tree-based ensemble method with leaf-wise growth

- Fast training, handles missing values natively
- Hyperparameter tuning: Optuna with 100 trials on GPU
- Best suited for: Tabular data with mixed feature types

SVR (Support Vector Regression)

- Kernel-based method (RBF = Radial Basis Function kernel)
- Maps data to higher dimensions for non-linear patterns
- Feature selection: Top 50 features by correlation
- Hyperparameter tuning: RandomizedSearchCV (50 iterations)

SARIMAX (Seasonal ARIMA with Exogenous Variables)

- Statistical time series model
- Captures seasonality (weekly patterns) and external drivers
- Order selection: Auto ARIMA via pmdarima
- Best suited for: Interpretable forecasts, price regression

---

3. KEY RESULTS

Performance Summary

Model	Accuracy	ROI	Sharpe Ratio
LightGBM	**70.31%**	**+16.86%**	**1.40**
SVR	67.87%	-13.85%	-1.15
SARIMAX	61.14%	-7.89%	-0.65

Key Takeaways:

- **LightGBM is the clear winner** - only model with positive ROI
- Sharpe ratio of 1.40 indicates good risk-adjusted returns (>1 is acceptable)
- SVR has high accuracy but predicts only “Down” - not useful for trading
- SARIMAX performs near-random for classification

Model Comparison Charts

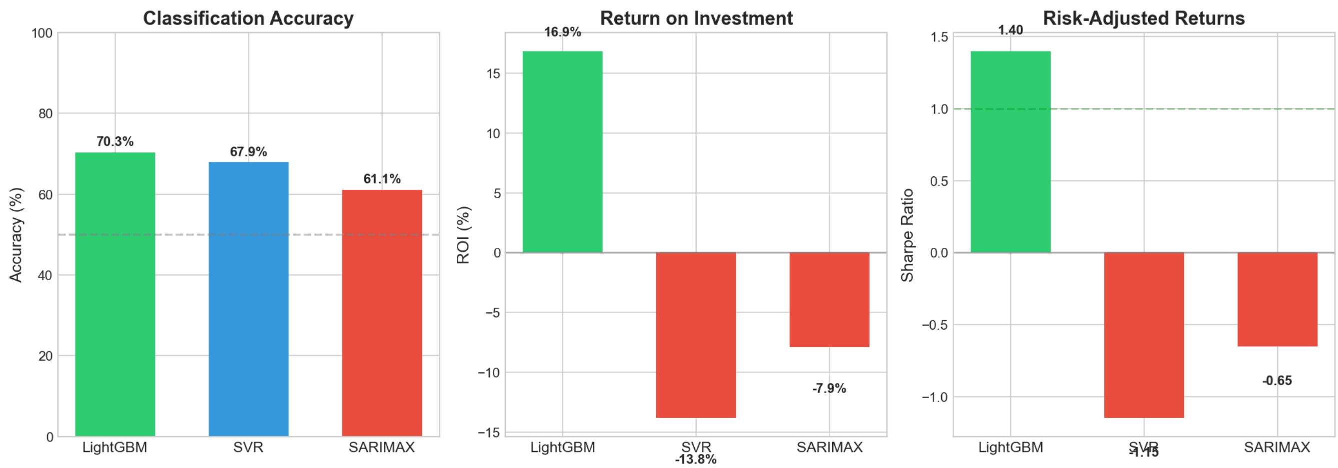

Figure 1: Model Comparison

---

4. LIGHTGBM DEEP DIVE

Why LightGBM Won:

- Captures non-linear feature interactions
- Handles the 27 features efficiently
- Optuna found optimal regularization (reg_lambda=3.5)

Top Predictive Features:

1. **price_position** (391) - Where price is relative to recent range
2. **pjm_load_pct_change** (383) - Daily load change
3. **Weekday** (369) - Day of week effect
4. **pjm_load** (361) - Absolute load level
5. **temperature** (319) - Weather impact

Interpretation:

- Load and temperature drive electricity demand
- Price position captures mean-reversion tendency
- Weekday captures business vs weekend patterns

Feature Importance Chart

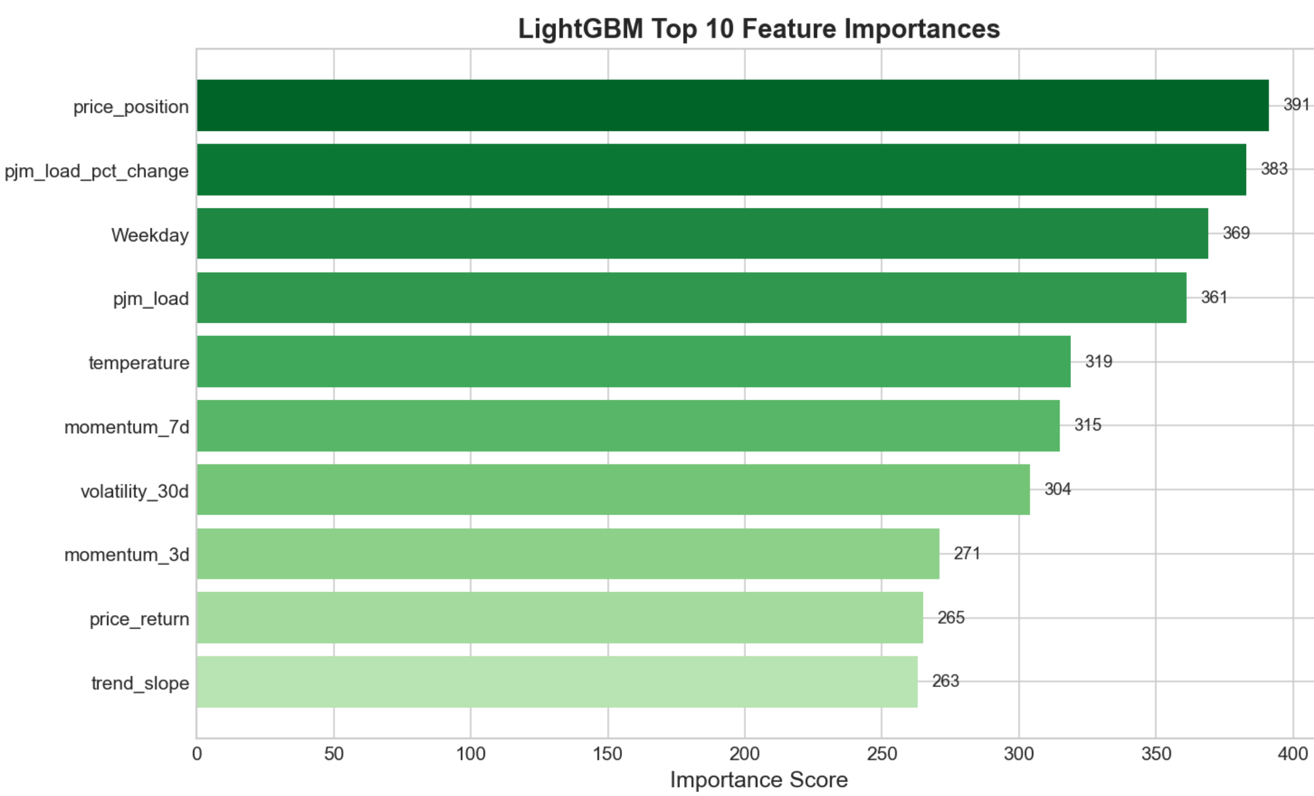

Figure 2: Feature Importance

---

5. TRADING PERFORMANCE

Strategy:

- Go LONG when model predicts "Up"

- Go SHORT when model predicts "Down"
- Trade every signal (1,189 total trades)

LightGBM Results:

- Win Rate: 37.93% (wins fewer trades but wins are larger)
- Average Return per Trade: +1.42%
- Total ROI: +16.86% over test period
- Sharpe Ratio: 1.40 (good risk-adjusted return)

Why SVR/SARIMAX Failed:

- Both predict mostly "Down" (majority class bias)
- Miss all upward price movements
- Negative ROI despite decent accuracy

---

6. CONFUSION MATRIX ANALYSIS

LightGBM Predictions:

	Pred Down	Pred Up
Actual Down	751 (93%)	56 (7%)
Actual Up	297 (78%)	85 (22%)

Interpretation:

- Very good at predicting "Down" (93% correct)
- Conservative with "Up" predictions (only 141 total)
- When it predicts "Up", it's right 60% of the time (85/141)
- Room for improvement: Better "Up" class detection

Confusion Matrix Visualization

---

7. MULTI-METRIC COMPARISON

Radar Chart - All Models

Summary Table

---

8. METHODOLOGY HIGHLIGHTS

Hyperparameter Tuning:

- **LightGBM**: Optuna TPE sampler, 100 trials, 5-fold TimeSeriesSplit
- **SVR**: RandomizedSearchCV, 50 iterations
- **SARIMAX**: AIC-based auto_arima order selection

Evaluation Metrics:

- **Accuracy**: Basic correctness measure
- **ROI**: Actual trading profitability
- **Sharpe Ratio**: Risk-adjusted returns (accounts for volatility)
- **Win Rate**: Percentage of profitable trades

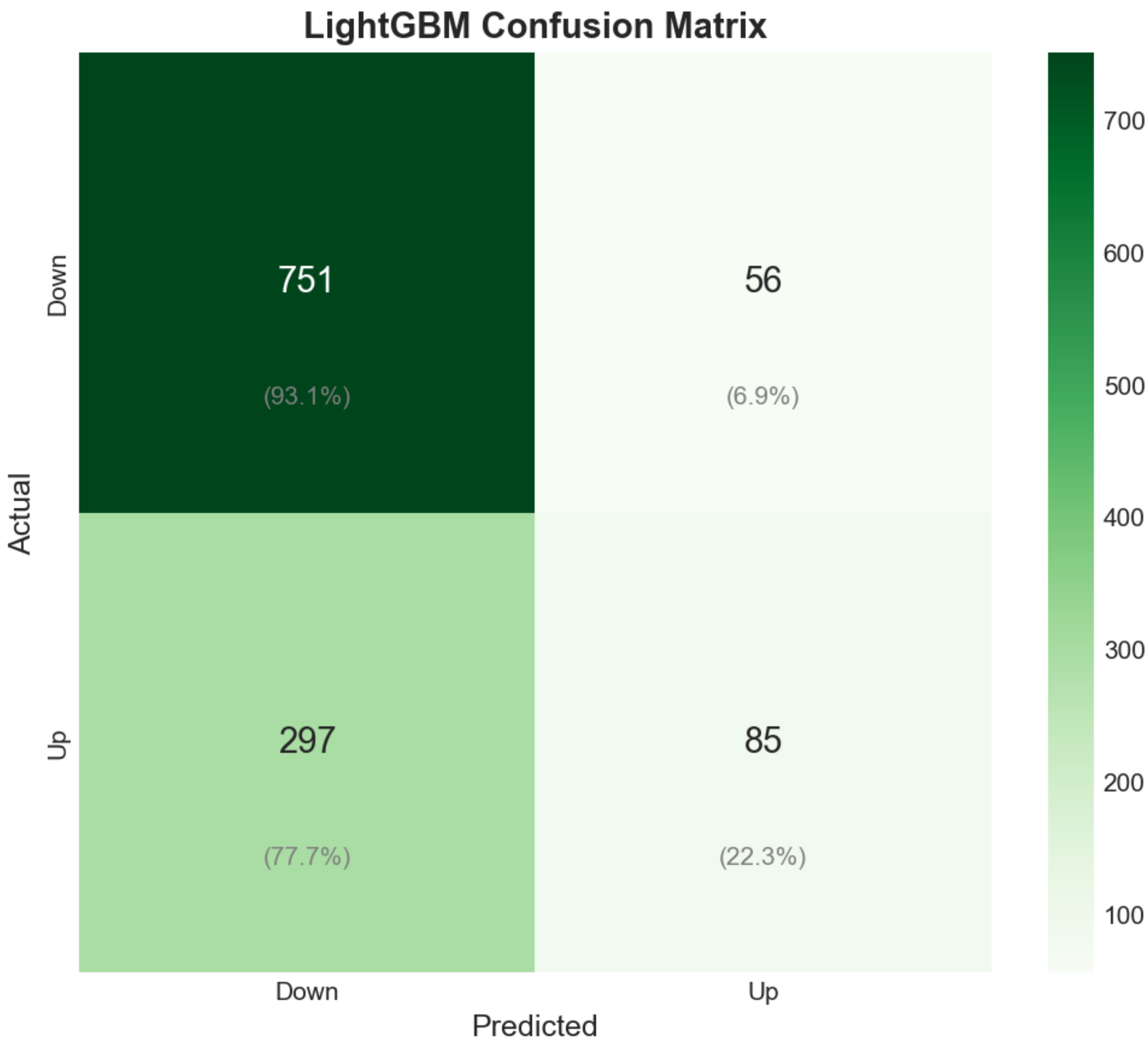

Figure 3: Confusion Matrix

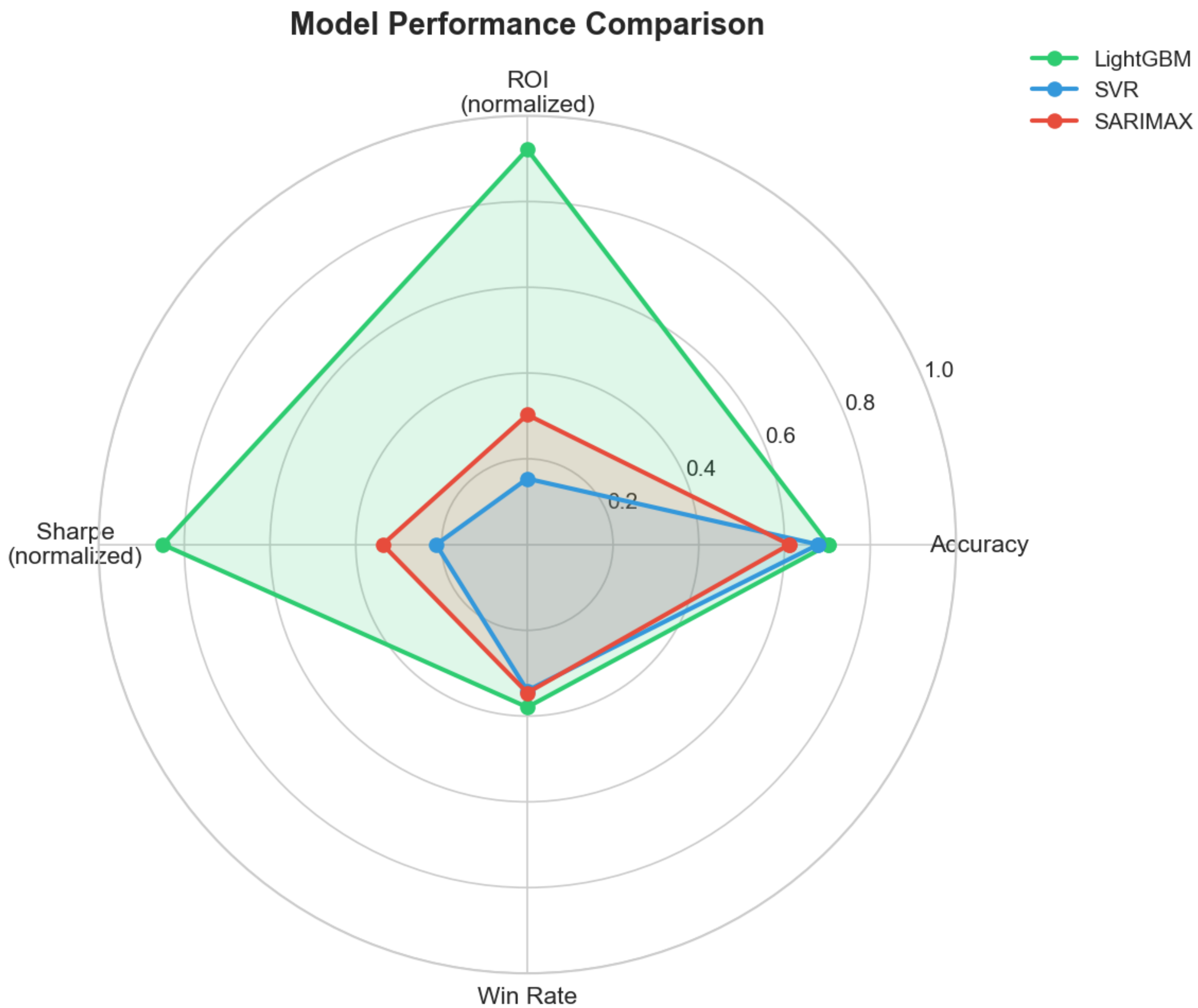

Figure 4: Radar Comparison

Model Performance Summary

Model	Accuracy	ROI	Sharpe	Win Rate	Best For
LightGBM	70.31%	+16.86%	1.40	37.93%	Trading
SVR	67.87%	-13.85%	-1.15	34.06%	Classification
SARIMAX	61.14%	-7.89%	-0.65	34.57%	Baseline

Figure 5: Summary Table

Why Sharpe Matters:

- A high-accuracy model can still lose money (SVR example)
- Sharpe > 1 means returns justify the risk
- LightGBM's 1.40 Sharpe is production-viable

---

9. LIMITATIONS & FUTURE WORK

Current Limitations:

- Class imbalance (68% Down, 32% Up) affects minority class
- No transaction costs or slippage in ROI calculation
- Single market (PJM) - may not generalize

Future Improvements:

1. **Class Imbalance**: SMOTE oversampling or focal loss
2. **Ensemble**: Combine LightGBM signals with SARIMAX price forecasts
3. **Feature Engineering**: Add more lagged features, Fourier terms
4. **Threshold Tuning**: Lower decision threshold for more "Up" signals

---

10. CONCLUSIONS

1. **LightGBM outperforms** traditional statistical (SARIMAX) and kernel (SVR) methods
2. **70.31% accuracy** with **+16.86% ROI** and **Sharpe 1.40**
3. **Key drivers**: Price position, load changes, temperature, weekday
4. **Production-ready**: Sharpe > 1 indicates viable trading strategy
5. **Next steps**: Address class imbalance, add ensemble methods

---

Q&A PREP - ANTICIPATED QUESTIONS

Q: Why not use deep learning (LSTM, Transformer)? A: LightGBM often outperforms DL on tabular data with <100k samples. DL shines with sequential patterns and large datasets.

Q: Is 70% accuracy good enough? A: For trading, ROI and Sharpe matter more than accuracy. Our 70% accuracy yields positive ROI, while SVR's 68% yields negative ROI.

Q: Why does SVR predict only "Down"? A: Class imbalance (68% Down). SVR's probability calibration pushes all predictions below 0.5 threshold.

Q: How would this work in production? A: Daily batch prediction before market open. Use prediction probability for position sizing. Implement stop-loss at 2% drawdown.

Q: What about overfitting? A: TimeSeriesSplit CV prevents look-ahead bias. Regularization (reg_lambda=3.5) prevents overfitting. Test set is completely held out.