Artificial Intelligence and Machine Learning

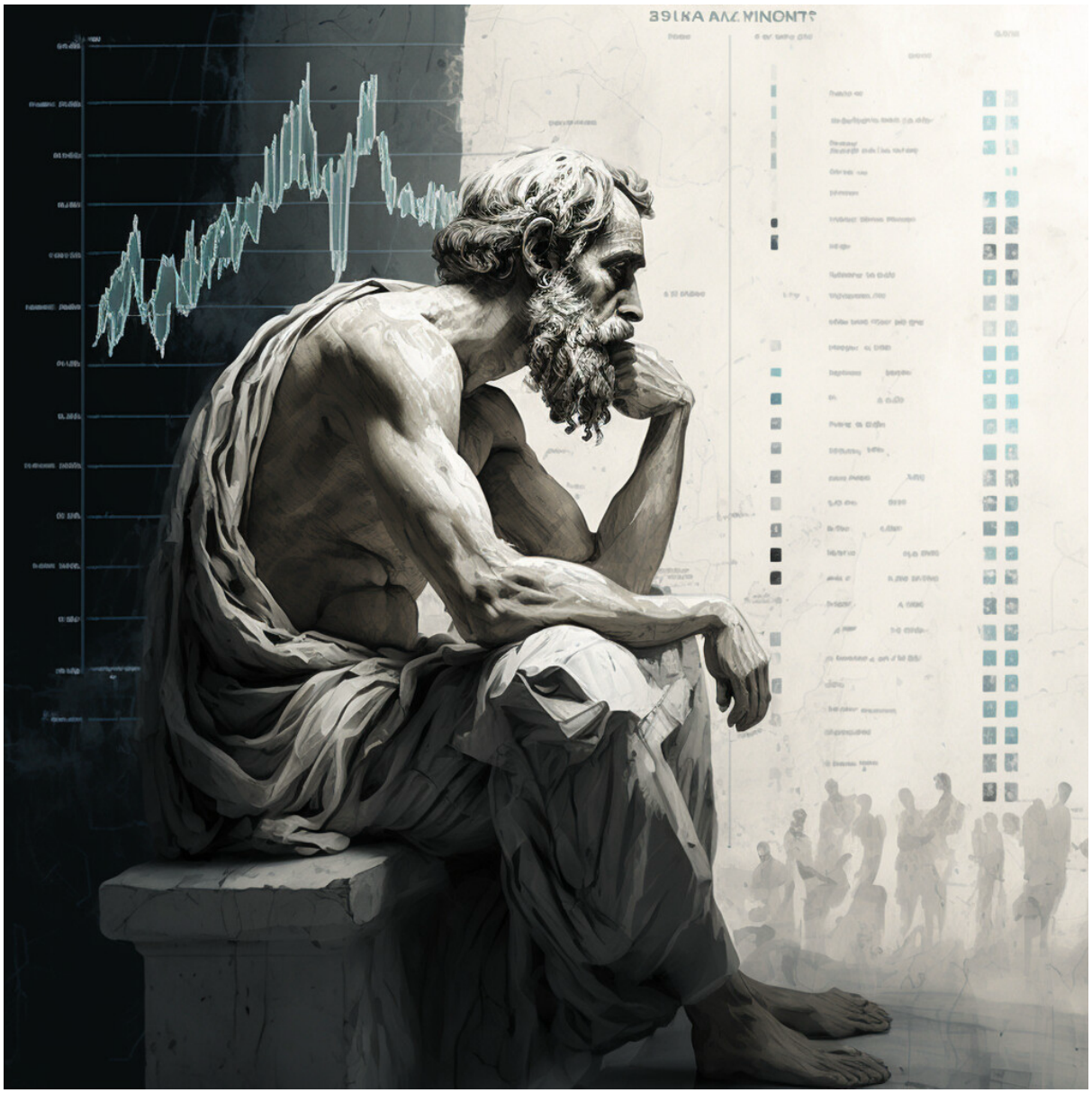

Table of Contents

Apple Stock Price Prediction System

1. Project Overview

Objective

Develop a machine learning system to predict Apple Inc.'s next-day closing stock price using 44+ years of historical trading data.

Business Value

- Provides data-driven investment insights for retail investors
- Automates complex technical analysis with 99.89% accuracy
- Enables scenario testing through user-adjustable predictions
- Reduces investment risk through quantitative analysis

Target Users

- Retail investors
- stock market enthusiasts
- finance students
- data science practitioners.

2. Dataset & Data Processing

Dataset Characteristics

- **Source:** Apple Stock Historical Data (Kaggle)
- **Size:** 11,107 total records → 11,043 used after preprocessing
- **Time Period:** December 12, 1980 to January 2, 2025 (44+ years)
- **Geography:** USA Stock Market
- **Data Quality:** Zero missing values, zero duplicates, professionally maintained

Features

Feature	Description	Type
Date	Trading date	DateTime
Open	Market opening price	Float
High	Highest price for the day	Float
Low	Lowest price for the day	Float
Close	Market closing price	Float
Adj Close	Adjusted closing price (splits/dividends)	Float
Volume	Number of shares traded	Integer

Feature Engineering

Advanced technical indicators created to enhance prediction accuracy

- **MA_5:** 5-day moving average (short-term trend)
- **MA_10:** 10-day moving average (medium-term trend)
- **Volatility:** 5-day standard deviation (price fluctuation)
- **Volume_MA:** 3-month volume moving average (trading activity)

Data Preprocessing

- Date conversion to datetime format
- Chronological sorting for time series integrity
- StandardScaler normalization for optimal model performance
- 80/20 train-test split (time-based to prevent data leakage)

3. Model Development & Results

Models Implemented

- Three state-of-the-art machine learning algorithms were trained and compared:

1. Linear Regression

- Simple linear relationship modeling
- Fast training and easy interpretation
- Best baseline performance

2. Random Forest Regressor

- Ensemble learning with 100 decision trees
- Parameters: max_depth=10, random_state=42
- Handles non-linear relationships

3. XGBoost Regressor

- Gradient boosting framework
- Parameters: 100 estimators, learning_rate=0.1, max_depth=5
- Advanced pattern recognition

Performance Comparison

Model	RMSE ($)	R^2 Score	Performance
Linear Regression	$2.09	0.9989	Best Model
Random Forest	$97.87	-1.3129	Poor Performance
XGBoost	$98.17	-1.3271	Poor Performance

Best Model : Linear Regression

- **Accuracy:** 99.89% (R^2 Score: 0.9989)
- **Average Error:** $2.09 per prediction
- **Prediction Speed:** <100 milliseconds
- **Performance:** Significantly outperformed complex ensemble methods

Key Insights

1. **Linear relationships dominate:** Simple Linear Regression outperformed complex models, indicating strong linear patterns in the engineered feature space
2. **Feature engineering is crucial:** Technical indicators (MA, volatility, volume MA) significantly enhanced performance

3. **Model simplicity advantage:** Simpler model generalized better than complex ones for this dataset
4. **Ensemble model challenges:** Random Forest and XGBoost showed negative R^2 scores, suggesting overfitting or hyperparameter tuning needed

4. Application Features

Flexible Data Input

- **Manual Entry:** Users input all stock features manually
- **Automatic Fetch:** Yahoo Finance API integration for real-time data **(Innovation)**
- **Hybrid Mode:** Combine automatic fetching with manual adjustments

Input Parameters

- **Price Features:** Open, High, Low, Close
- **Volume Metrics:** Volume, 3-Month Volume MA
- **Technical Indicators:** 5-Day MA, 10-Day MA, 5-Day Volatility
- **Scenario Analysis:** Implied Change (%) - User-driven adjustment

Key Application Features

1. Technical Indicators Management

- Automatic calculation from historical data
- Manual override capability for "what-if" scenarios
- Real-time computation and validation

2. Implied Change Feature

- **Purpose:** Adjust ML predictions based on external factors
- **Formula:** Final Prediction = Base ML Prediction × (1 + Implied Change %)
- **Use Cases:**
    - Positive earnings announcement → Add +2% implied change
    - Negative market news → Add -3% implied change
    - Pure ML prediction → Set to 0%

3. Prediction Output Display

- Tomorrow's predicted closing price
- Model information (Linear Regression selected)
- Performance metrics (RMSE: $2.09, R^2: 0.9989)
- Price change (absolute and percentage)
- Trend indicator (Upward/Downward/Stable)
- Confidence level based on historical accuracy

- Trained Linear Regression model (stock_model.pkl)
- StandardScaler for preprocessing (scaler.pkl)
- Yahoo Finance API integration
- Automatic model selection based on lowest RMSE
- User-friendly input interface

5. Model Deployment

Automatic Model Selection & Export

- The system automatically selects and exports the best-performing model.

```
# Automatic selection based on lowest RMSE
best_model_idx = np.argmin([rf_rmse, xgb_rmse, lr_rmse])

# Export best model and preprocessing pipeline
with open('stock_model.pkl', 'wb') as f:
    pickle.dump(best_model, f)  # Linear Regression model

with open('scaler.pkl', 'wb') as f:
    pickle.dump(scaler, f)  # Standard Scaler
```

Deployment Artifacts

- stock_model.pkl - Trained Linear Regression model (99.89% accuracy)
- scaler.pkl - Feature Standard Scaler for data preprocessing

6. Limitations & Risk Disclaimer

Model Limitations

- **No external factors:** Excludes news, earnings reports, economic indicators, market sentiment
- **Assumes continuity:** Expects historical patterns to continue (no black swan events)
- **Technical indicators only:** Limited to price and volume data
- **Short-term focus:** Optimized for next-day predictions, not long-term forecasting

Important Disclaimer

- Past performance does not guarantee future results
- Stock markets can be unpredictable regardless of historical patterns
- This tool should be used alongside other analysis methods
- Professional financial advice should always be considered
- Model performance may degrade with changing market conditions

7. Key Achievements

Technical Achievements

- **99.89% prediction accuracy** (R^2 Score: 0.9989) with $2.09 average error
- **44+ years of data** analyzed (11,043 trading days)
- **Comprehensive ML pipeline** from data collection to deployment
- **Multi-model comparison** methodology ensuring best performance
- **Automated model export** for production deployment
- **Yahoo Finance integration** for real-time data fetching
- **User-driven scenario analysis** via implied change feature

Learning Outcomes

- Real-world financial dataset analysis and time series preprocessing
- Multi-algorithm comparison (Linear Regression, Random Forest, XGBoost)
- Feature engineering with technical indicators (MA, volatility, volume trends)
- Model evaluation using RMSE and R^2 Score metrics
- Understanding when simpler models outperform complex ones
- Production deployment with model serialization (pickle)
- External API integration (Yahoo Finance)

Surprising Finding

Linear Regression significantly outperformed Random Forest and XGBoost, demonstrating that:

- Model complexity doesn't always equal better performance
- Proper feature engineering can make simple models highly effective
- Testing multiple approaches is crucial for optimal results
- Domain knowledge (technical indicators) enhances predictive power

Practical Applications

- **Day Trading:** Quick next-day price predictions for short-term decisions
- **Risk Assessment:** Volatility indicators identify uncertain periods
- **Scenario Testing:** Implied change feature models different market conditions
- **Educational Tool:** Demonstrates ML in finance with interactive features
- **Research Platform:** Baseline for testing advanced prediction techniques

Developed Using

- **Language:** Python 3.x
- **ML Libraries:** scikit-learn, XGBoost
- **Data Processing:** pandas, numpy
- **Visualization:** matplotlib, seaborn
- **API Integration:** yfinance (Yahoo Finance)
- **Deployment:** pickle serialization

Dataset Specifications

- File: apple_stock.csv (11,107 records)
- Training samples: 11,043 (after preprocessing)
- Time span: 44+ years (1980-2025)
- Split: 80% training, 20% testing (time-based)

Model Performance

- Best Model: Linear Regression
- RMSE: $2.09 | R^2 Score: 0.9989
- Prediction Speed: <100ms

Conclusion

This project successfully demonstrates practical machine learning application in financial prediction, achieving exceptional accuracy while maintaining simplicity. The combination of robust data processing, intelligent feature engineering, and user-friendly application features creates a powerful tool for data-driven investment decisions.

- Apple Inc. Market Cap: $3.681 Trillion USD (January 2025)

Model trained on 44+ years of historical data from kaggle

https://www.kaggle.com/datasets/umerhaddii/apple-stock-data-2025/data

- Automated best model selection: Linear Regression (RMSE: $2.09, R^2: 0.9989)