Directional Forecasting Cryptocurrencies Project Overview Report

jiangqin.ma@gmail.com

1 INTRODUCTION

Cryptocurrency markets operate around the clock, presenting unique challenges and opportunities for traders and analysts. In this project, we undertake an exploratory data analysis (EDA) of a high-frequency dataset containing minute-by-minute OHLCV (Open, High, Low, Close, Volume) data for a specific cryptocurrency. The data captures essential details of price movements and trading volumes, providing a comprehensive view of market activity at a granular level (GenAI.Labs, 2024).

Beyond EDA, we aim to build predictive models capable of forecasting short-term price movements. This challenge offers a unique opportunity to delve into high-frequency trading insights within the volatile and fast-paced cryptocurrency market. It encourages innovative feature engineering, the application of cutting-edge machine learning techniques, and provides a realistic simulation of market impact.

The primary objective of this analysis is to understand the dataset's structure, uncover hidden trends, and identify anomalies that could impact predictive modeling. This EDA will serve as a foundation for developing robust models to predict short-term price movements, thereby contributing valuable insights into the dynamic and volatile cryptocurrency market.

2 EXPLORATORY DATA ANALYSIS (EDA)

2.1 Dataset Overview

The dataset provided for this project contains historical minute-by-minute OHLCV (Open, High, Low, Close, Volume) data for a specific cryptocurrency. It covers a time span from May 4, 2018, at 22:01 to May 17, 2022, at 19:58, comprising a total of 2,122,438 records across 11 columns. Each record represents one minute of trading activity, capturing essential metrics such as price movement, trading volume, and the number of trades, providing a detailed view of the cryptocurrency's market behavior.

The dataset includes columns like `timestamp`, which records the time of each data point, and price-related metrics (`open`, `high`, `low`, and `close`), which reflect the cryptocurrency's price changes during the minute. Additionally, it features `volume`, indicating the number of units traded, and `quote_asset_volume`, representing the total value of traded assets in USDT. Other columns, such as `number_of_trades`, `taker_buy_base_volume`, and `taker_buy_quote_volume`, provide insights into trading activity and market participation. The `target` column identifies the direction of the price movement in the next minute, where a value of 1 signifies an upward movement and 0 indicates no change or a downward movement.

2.2 Target Distribution

The target variable represents the direction of the cryptocurrency price movement in the next minute: 1 indicates an upward price movement, while 0 signifies no change or a downward movement. The dataset is reasonably balanced between the two classes, ensuring fair model performance across both outcomes.

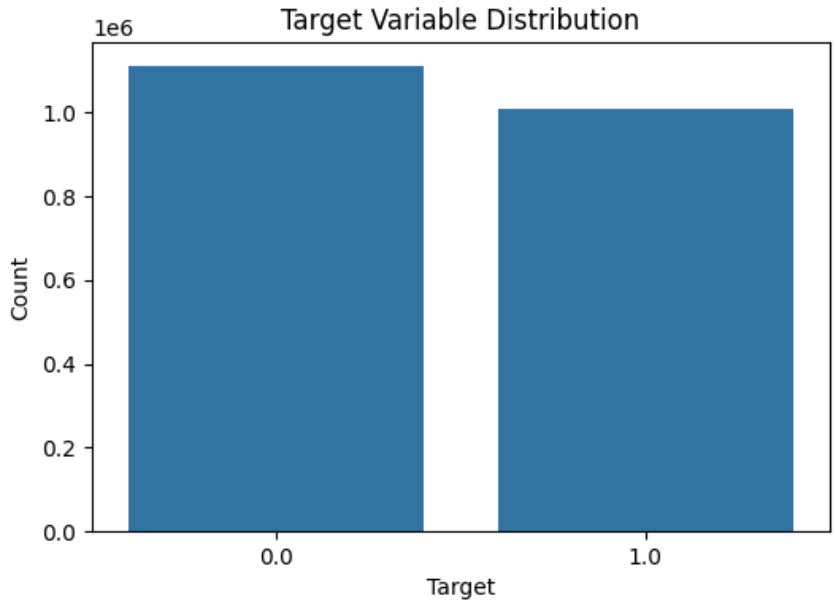

Figure 1—Target Variable Distribution

2.3 Correlation Analysis

A correlation analysis was conducted to identify relationships between features and the target variable. The results reveal minimal correlation between the `close`

price and the target variable. However, features such as `trading_volume` and the `number_of_trades` exhibit slight positive correlations with the target, suggesting they may hold some predictive value.

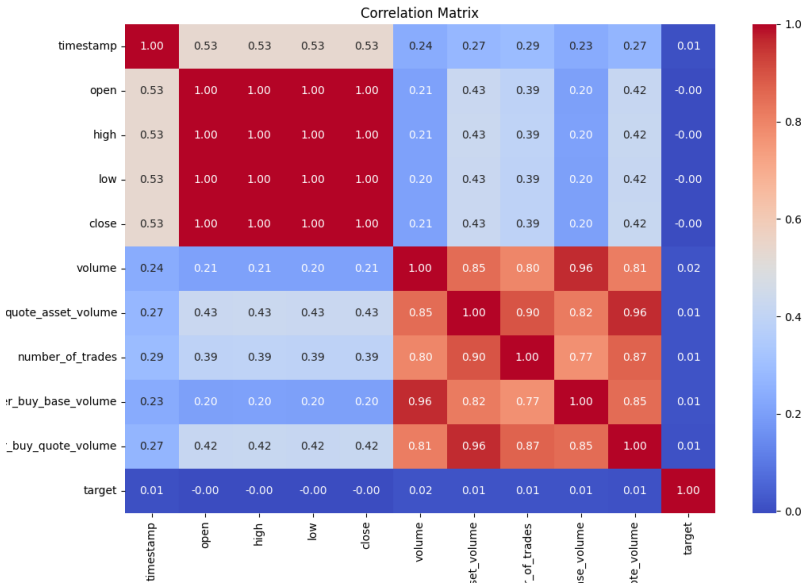

Figure 2—Correlation Matrix

2.4 Close Price Over Time

The time series of the `close` price reveals the overall price trends and volatility in the cryptocurrency market. Significant spikes and drops in the price indicate periods of high volatility, which are characteristic of cryptocurrency markets.

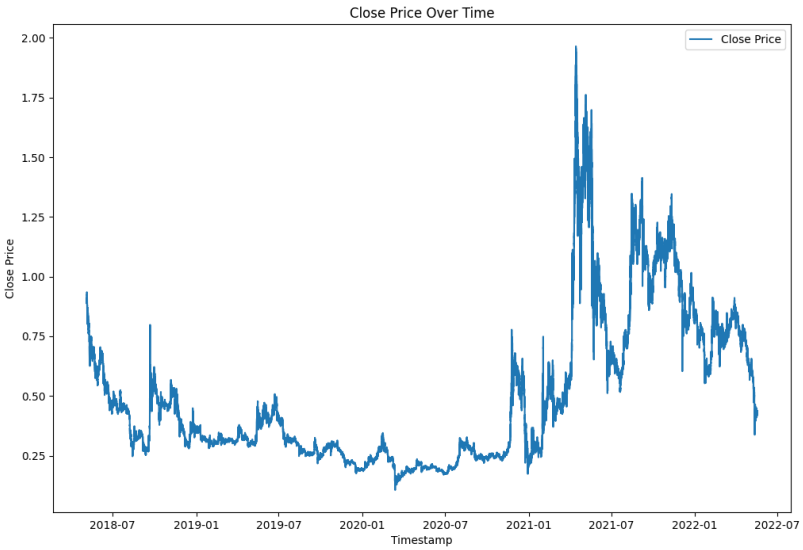

Figure 3—Close Price Over Time

2.5 Trading Volume Over Time

The `trading_volume` provides insights into market activity, with noticeable spikes pointing to periods of heightened trading activity that could signal increased market interest or external events.

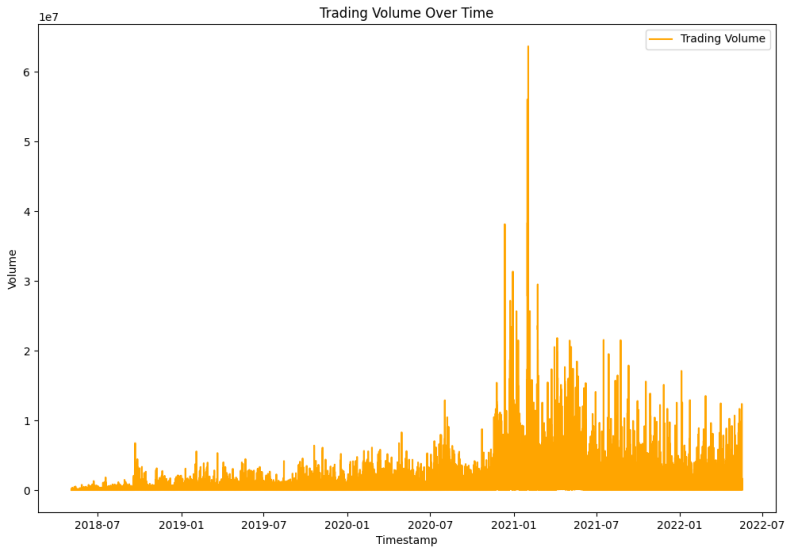

Figure 4—Trading Volume Over Time

2.6 Close Price by Target

The boxplot below shows how the `close` price varies with the target values. Both classes exhibit similar distributions, but higher prices appear slightly more common in upward movements (`target = 1`).

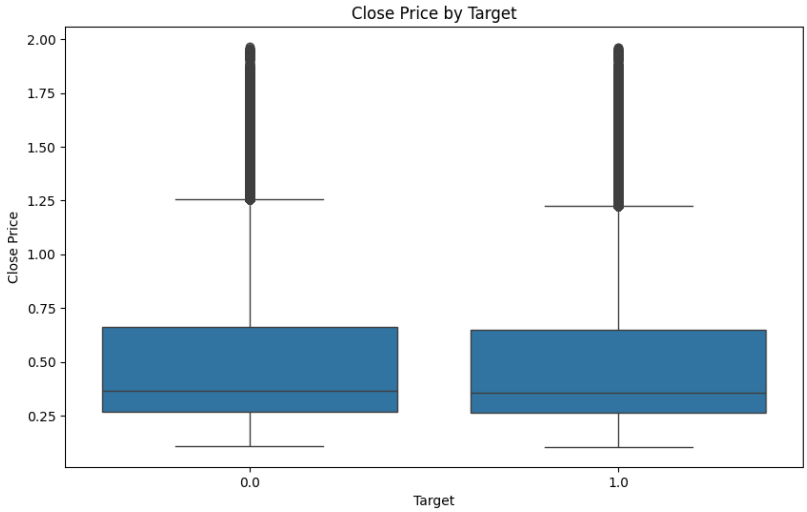

Figure 5—Close Price by Target

2.7 Volume Outliers

A boxplot of the `trading_volume` highlights the presence of extreme values. While most trading volumes fall within a reasonable range, the outliers could significantly impact model performance if not appropriately handled during preprocessing.

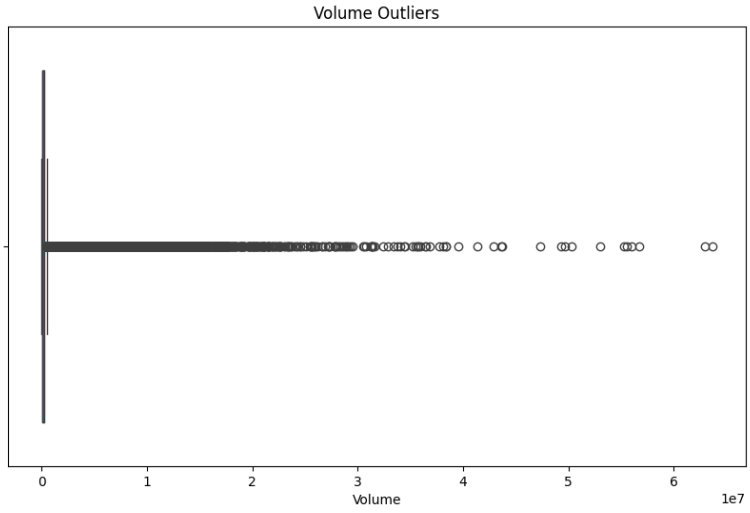

Figure 6—Volume Outliers

SUMMARY OF EDA

The EDA highlights several important aspects of the dataset. The balanced target variable distribution is promising for fair model evaluation. While individual feature correlations with the target are minimal, trading activity features like `volume` and `number_of_trades` may hold predictive power. The time series plots underscore the volatility and activity spikes characteristic of cryptocurrency markets. Additionally, the strong relationships between volume-related features provide opportunities for further feature engineering. These insights lay the groundwork for building robust predictive models.

3 METHODS

3.1 Overview

In this project, I employ three distinct modeling approaches to predict the next-minute price movement of a cryptocurrency based on historical data. The methods are as follows:

1. **SARIMAX (Seasonal AutoRegressive Integrated Moving Average with eXogenous regressors):** SARIMAX is a time-series forecasting model that incorporates exogenous variables to improve prediction accuracy. We use historical price features such as returns, volume, and volatility to predict future price movements.
2. **Hybrid SARIMAX and XGBoost Model:** This hybrid approach combines the strengths of SARIMAX for time-series prediction and XGBoost for capturing complex nonlinear relationships. SARIMAX outputs are used as features in XGBoost, creating a pipeline that leverages both statistical and machine learning methods.
3. **Spark Triple Ensemble Model:** This ensemble model integrates:
   - Gradient Boosted Trees (GBT)
   - Random Forest (RF)
   - Long Short-Term Memory (LSTM) Neural Networks

   The ensemble model is implemented in Apache Spark for scalability, making it suitable for high-frequency, large-scale data processing.

3.2 SARIMAX

The SARIMAX model is trained using historical time-series data, with exogenous variables derived from engineered features. The following steps are performed:

1. **Technical Indicator Calculation:** We calculate various technical indicators, including:
   - **Returns:** Percentage change in closing price.
   - **RSI (Relative Strength Index):** Measures the magnitude of recent price changes to evaluate overbought or oversold conditions (Investopedia, n.d.(b)).
   - **MACD (Moving Average Convergence Divergence):** Captures momentum by calculating the difference between fast and slow exponential moving averages (Investopedia, n.d.(c)).
   - **Rolling Averages:** Includes the 5-period rolling mean of closing prices and volume.

- **Price Range:** Difference between high and low prices normalized by the opening price.
- **Time Features:** Hour of the day and day of the week to account for temporal effects.

2. **Missing Value Handling:** Missing values in the dataset are addressed using forward and backward filling methods. Any remaining missing values are replaced with zeros to ensure model compatibility.
3. **Feature Preparation:** Exogenous features are selected and standardized using a 'StandardScaler'. The selected features include technical indicators and time features.
4. **Model Training:** The SARIMAX model is trained with the following configuration:
   - **Order:** $(1, 0, 0)$ — a simple AutoRegressive model of order 1 is employed.
   - **Exogenous Variables:** The standardized features prepared in the earlier step are used as exogenous inputs.
   - **Stationarity:** The model allows non-stationary data by setting `enforce_stationarity=False`.

This approach ensures that the SARIMAX model captures both temporal dependencies and the impact of exogenous variables on the target variable. Predictions are made for each future timestamp using the fitted model, and directional movement (up or not up) is determined by comparing the predicted price with the current price.

3.3 Hybrid SARIMAX and XGBoost

The Hybrid SARIMAX and XGBoost model combines the strengths of statistical modeling and machine learning to predict the next-minute price direction. The process involves:

1. **Feature Engineering:**
   - **Returns:** Percentage change in closing price.
   - **Momentum Indicators:** RSI, MACD, and ATR are calculated using shifted historical data to avoid label leakage. ATR (Average True Range) measures market volatility by averaging the true range over a specified number of periods. The true range is calculated as the greatest of:
     (a) The current high minus the current low,
     (b) The absolute value of the current high minus the previous close,
     (c) The absolute value of the current low minus the previous close.

ATR is particularly useful for assessing volatility and risk management in trading strategies (Investopedia, n.d.(a)).

- **Volume-Based Indicators:** Volume ratio and moving average are used to capture short-term trading activity.
- **Time Features:** Hour of the day and day of the week are included to account for temporal effects.

2. **SARIMAX Model:**
   - SARIMAX is trained on the most recent 50,000 rows of the training dataset.
   - Exogenous features (e.g., momentum and volume indicators) are standardized and included in the model.
   - Hyperparameters such as (p, d, q) and seasonal orders are tuned using a grid search over predefined parameter ranges.
3. **XGBoost Model:**
   - XGBoost is trained on the entire training dataset with standardized features and the target variable.
   - The model's hyperparameters, including learning rate, maximum depth, number of estimators, and subsample ratio, are fine-tuned through grid search.
   - The tree-based model captures complex nonlinear relationships in the data.
4. **Fine-Tuning Process:**
   - **Grid Search:** Both SARIMAX and XGBoost models are fine-tuned using a grid search approach. For SARIMAX, combinations of (p, d, q) orders and seasonal orders are explored. For XGBoost, parameters such as learning rate, maximum depth, number of trees, and subsample ratios are evaluated.
   - **Validation Split:** A temporal validation split is used to ensure that the model generalizes to unseen future data. The training set uses the most recent data for SARIMAX and the entire dataset for XGBoost.
   - **Evaluation Metric:** The Macro-Averaged F1 Score is used to evaluate model performance during fine-tuning, ensuring balanced performance across classes.
   - **Ensemble Weight Optimization:** A weight parameter is introduced to balance the contributions of SARIMAX and XGBoost predictions. The optimal weight is selected based on validation performance.
5. **Hybrid Prediction:**
   - Predictions from SARIMAX are based on directional price changes derived from its forecast.
   - Predictions from XGBoost are derived from the model's output probabilities.

- The final hybrid prediction is calculated as:

  $$\text{Final Prediction} = w \cdot \text{SARIMAX Predictions} + (1 - w) \cdot \text{XGBoost Predictions}$$

  where w is the optimized ensemble weight.

The hybrid model effectively combines the temporal insights from SARIMAX with the nonlinear predictive power of XGBoost, resulting in a robust approach to predicting cryptocurrency price directions.

3.4 Spark Triple Ensemble Model

The Spark Triple Ensemble Model leverages Apache Spark for distributed computation and includes three sub-models:

- **GBT and RF:** These tree-based models are trained using scaled features such as returns, RSI, MACD, and Bollinger Bands.
- **LSTM:** The LSTM model processes sequential data to capture temporal dependencies in the price movements.

3.4.1 *Feature Engineering*

Key features engineered include:

- **Technical Indicators:** RSI, MACD, Bollinger Bands.
- **Volume Metrics:** Volume ratio.
- **Target Variable:** A binary target indicating whether the price moves up (1) or stays the same/moves down (0) in the next minute.

3.4.2 *Training Procedure*

1. Datasets have training and testing sets.
2. Each model is trained on the training set:
   - **GBT and RF:** Pipelines include feature scaling and model training.
   - **LSTM:** Sequential data is prepared using a sliding window approach.

3.4.3 *Prediction*

Ensemble predictions are calculated as:

$$\text{Final Prediction} = 0.5 \cdot (\text{GBT Predictions}) + 0.5 \cdot (\text{RF Predictions})$$

The LSTM predictions are used for validation and ensemble refinement.

4 EVALUATION

4.1 Evaluation Metric

The model's performance is evaluated using the **Macro-Averaged F1 Score**, which assesses the balance between precision and recall for each class (up and not up).

4.1.1 *Formula*

The F1 score for each class c is calculated as:

$$F1_c = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

The Macro-Averaged F1 Score is:

$$F1_{\text{macro}} = \frac{1}{C} \sum_{c=1}^{C} F1_c$$

Where $C = 2$, representing the two classes (GenAI.Labs, 2024).

5 RESULTS

5.1 Model Comparison

The performance of each model is compared using the Macro-Averaged F1 Score on the test set. The evaluation focuses on:

- **SARIMAX:** Evaluates the model's ability to capture time-series trends.
- **Hybrid SARIMAX and XGBoost:** Measures the improvement from incorporating nonlinear modeling.
- **Spark Triple Ensemble Model:** Assesses the robustness of the ensemble approach.

5.2 Results Interpretation

High F1 scores across both classes indicate that the model balances false positives and false negatives effectively. Ensemble methods are expected to outperform single models due to their ability to leverage diverse modeling techniques.

This section presents the performance of various regression models (OLS, Best Subset, Stepwise AIC, Ridge, LASSO, PCR, and PLS) based on training and testing errors, variances, coefficients, and plots. The key metrics used to evaluate the models are the Mean Squared Error (MSE) and variance for both training and testing datasets.

5.3 Training and Testing Errors

The models' performance, in terms of Training and Testing MSE and Variance, is summarized in Table 1 and Table 2. The following observations can be made:

Best Subset achieves the lowest testing MSE (0.0028) but has a slightly higher training error (0.0315), indicating a strong ability to generalize.

LASSO has one of the lowest testing errors (0.0032) and testing variances (0.0033), demonstrating the benefit of regularization for high-dimensional data.

OLS, Ridge, PCR, and PLS perform similarly, with marginal differences in training and testing errors and variances. These models have lower complexity but slightly higher testing errors than the regularized methods.

Stepwise AIC exhibits a relatively higher testing error, suggesting potential overfitting, as it selects a more complex model than Ridge or LASSO.

These results are presented in the following tables:

Model	**Training Error**	**Testing Error**	**Training Variance**	**Testing Variance**
OLS	0.0293	0.0088	0.0294	0.0090
Best Subset	0.0315	0.0028	0.0316	0.0028
Stepwise AIC	0.0295	0.0090	0.0296	0.0092
Ridge	0.0293	0.0089	0.0294	0.0091
LASSO	0.0309	0.0032	0.0310	0.0033
PCR	0.0293	0.0088	0.0294	0.0090
PLS	0.0293	0.0088	0.0294	0.0090

Table 1—Training and Testing Errors, Variance for Different Models

Model	Training Error	Testing Error	Training Variance	Testing Variance
OLS	0.0266	0.0531	0.0267	0.0529
Best Subset	0.0283	0.0440	0.0284	0.0440
Stepwise AIC	0.0270	0.0519	0.0271	0.0518
Ridge	0.0266	0.0536	0.0267	0.0535
LASSO	0.0275	0.0414	0.0276	0.0414
PCR	0.0267	0.0501	0.0268	0.0501
PLS	0.0266	0.0514	0.0267	0.0514

Table 2—Average Training and Testing Errors, Variance for Different Models in Part (e) Using Monte Carlo Cross-Validation algorithm

5.4 Coefficients for Each Model

The coefficients estimated for each model are shown in Table 3. Several patterns emerge:

OLS, Ridge, PCR, and PLS have similar coefficient estimates, indicating that regularization does not significantly alter the direction or magnitude of the coefficients.

LASSO shrinks some coefficients to zero (e.g., weight and height), highlighting its effectiveness in feature selection by eliminating predictors that do not contribute much to the model.

Best Subset selects a different set of coefficients, suggesting a different combination of predictors, but the model exhibits similar general patterns to the OLS and Ridge models.

Attribute	OLS	Best Subset	Stepwise AIC	Ridge	LASSO	PCA	PLS
Intercept	12.3905	11.2034	12.5728	12.6095	11.2737	-3.1827	12.3905
siri	0.8834	0.9044	0.8842	0.8826	0.9040	0.3271	0.8834
density	-9.9981	-9.2427	-10.2111	-10.2017	-9.4059	-0.0007	-9.9981
age	-0.0007	0.0000	0.0000	-0.0007	-0.0004	-0.0124	-0.0007
weight	0.0115	0.0000	0.011021	0.0117	0.0000	0.2152	0.0115
height	-0.0008	0.0000	0.0000	-0.0008	0.0000	0.1482	-0.0008
adipos	-0.0189	0.0000	-0.0160	-0.0191	0.0000	-0.0084	-0.0189
free	-0.0133	0.0000	-0.0125	-0.0135	0.0000	-0.3361	-0.0133
neck	-0.0006	0.0000	0.0000	-0.0006	0.0000	0.0021	-0.0006
chest	0.0025	0.0000	0.0000	0.0026	0.0000	0.0064	0.0025
abdom	0.0007	0.0000	0.0000	0.0008	0.0000	0.1146	0.0007
hip	-0.0036	0.0000	0.0000	-0.0036	0.0000	0.0181	-0.0036
thigh	0.0146	0.0099	0.0131	0.0146	0.0070	0.0192	0.0146
knee	-0.0262	-0.0245	-0.0274	-0.0261	-0.0136	0.0221	-0.0262
ankle	0.0033	0.0000	0.0000	0.0032	0.0000	0.0089	0.0033
biceps	-0.0172	0.0000	-0.0172	-0.0172	-0.0089	0.0105	-0.0172
forearm	0.0238	0.0000	0.0256	0.0239	0.0133	0.0106	0.0238
wrist	0.0327	0.0289	0.0295	0.0328	0.0172	0.0006	0.0327

Table 3—Coefficients of the models for different attributes

5.5 Residuals vs Fitted Plots

Residual plots for OLS, Best Subset, and Stepwise AIC models indicate how well these models fit the data:

OLS: The residuals in Figure 5 are mostly centered around zero, but outliers, such as points 33, 169, and 182, exert influence.

Figure 7— Residual plots for OLS

Best Subset: Figure 6 shows better residual distribution, with fewer extreme residuals compared to OLS.

Stepwise AIC: The residuals in Figure 7 exhibit similar behavior to OLS, though the model appears slightly overfitted due to higher testing errors.

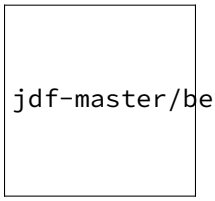

Figure 8— Residual plots for Best Subset

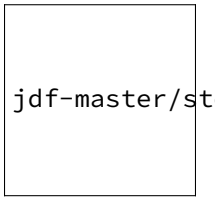

Figure 9— Residual plots for Stepwise AIC

5.6 Regularization Coefficients Paths

The Ridge coefficient path in Figure 8 indicates that the Ridge model stabilizes the coefficients and prevents overfitting by shrinking them as increases.

Figure 10— Ridge Coefficients Paths

LASSO performs feature selection by shrinking some coefficients to zero, making it effective for reducing model complexity.

Figure 11— LASSO Coefficients Paths

5.7 Dimensionality Reduction Validation

The dimensionality reduction techniques (PCA, PCR, and PLS) are evaluated using scree plots and validation curves:

Most variance is captured by the first few principal components, with diminishing returns after the third component.

Figure 12— PCA Scree Plot

MSE decreases sharply with the first few components and then stabilizes, indicating that only a subset of components is required for optimal performance.

Figure 13— PCR Validation Plot

Figure 14— PLS Validation Plot

6 FINDINGS

This section interprets the results and provides key insights into the performance of the models based on the training/testing errors, variances, coefficients, and visual analysis.

Best Subset and LASSO models show the best generalization with lower testing errors, effectively selecting relevant predictors. OLS, Ridge, PCR, and PLS perform similarly but with slightly higher errors, reflecting their more straightforward approaches.

LASSO improves interpretability by setting some coefficients to zero (e.g., weight, height), highlighting the most important predictors. Best Subset identifies unique predictor combinations, capturing relationships missed by simpler models.

Ridge and LASSO reduce overfitting through regularization. Ridge stabilizes coefficients as increases, while LASSO simplifies models with feature selection, making it more interpretable.

PCR and PLS show that dimensionality reduction reduces model complexity without sacrificing performance. PLS outperforms PCR by better capturing the relationship between predictors and the response, making it more efficient.

Best Subset and LASSO are recommended for generalization and feature selection. Ridge is suitable for controlling multicollinearity, while PLS is optimal for dimensionality reduction.

7 APPENDICES

8 REFERENCES

1. GenAI.Labs (2024). *Directional Forecasting Cryptocurrencies*. https://kaggle.com/competitions/directional-forecasting-cryptocurrencies. Kaggle.
2. Investopedia (n.d.[a]). *Average True Range (ATR) Indicator Explained With Formula*. https://www.investopedia.com/terms/a/atr.asp. Accessed: 2024-01-05.
3. Investopedia (n.d.[b]). "Relative Strength Index (RSI) Indicator Explained With Formula". In: *Investopedia Technical Analysis Resources*. Online: Investopedia. URL: https://www.investopedia.com/terms/r/rsi.asp.
4. Investopedia (n.d.[c]). "What Is MACD?" In: *Investopedia Technical Analysis Resources*. Online: Investopedia. URL: https://www.investopedia.com/terms/m/macd.asp.