[EM 1414] MODELS AND TECHNOLOGIES FOR THE FINANCIAL INDUSTRY

Predicting Crypto Movements & Optimizing Portfolios with Machine Learning

14 MARCH 2025
ANTONELLA FACINI 900455
GIORGIO BERNACCHI MONTI 886263

Index

Introduction

Data Preprocessing & EDA

Feature Selection

Linear & Polynomial Regression

RNN - LSTM

Random Forest

Dynamic Portfolio Allocation

Conclusion

Research Question

How effective are **machine learning models** in predicting **cryptos** prices and **movements**?

How can these **predictions** be leveraged for **portfolio management strategies**?

Introduction

Source

- **Yahoo Finance**
- Python's *yfinance* library was used
- Source: https://finance.yahoo.com/

Data

- **Focus:**
  - BTC - USD
  - ETH
  - USDT - USD
  - DOGE
- **Time span:** (start) 2018 - (end) 2024
- **Download:**
  - Daily Observations
  - O, H, L, Adjusted Close

Methodology

Classification Task:

- Predict if tomorrow's **% return** will be **Up, Down, or Neutral.**
- Models Used:
  - **Random Forest:** Tuned with Bayesian Optimization.
  - **LSTM:** Used to capture time series patterns.
- **Portfolio Strategy:** Allocations based on prediction confidence.

Key Variables (1)

Feature	Description
Date	Represents the date for which the data point is recorded. Helps analyze the temporal progression of cryptocurrency prices.
Open	The price of the cryptocurrency at the beginning of the trading day. Provides insight into market sentiment at the start of the trading session.
High	The highest price reached during the trading day. A key indicator of bullish behavior, showing the peak price achieved during the trading session.
Low	The lowest price reached during the trading day. Helps assess market corrections or dips, evaluating risk levels in a given timeframe.
Adj. Close	The adjusted closing price, accounting for unusual market events. Ensures accuracy even though less relevant for cryptocurrencies.
Volume	The total number of cryptocurrency units traded during the day. Reflects liquidity and market interest, with higher volume indicating activity.

Key Variables (2)

Indicator	Description
EMAF (Fast EMA)	A very short-period EMA, highly sensitive to recent price movements, used to detect immediate trend changes.
EMAM (Medium-Term EMA)	Highlights medium-term trends by smoothing price fluctuations.
EMAS (Short-Term EMA)	Reacts quickly to short-term price changes, identifying short-term trends.
Middle Band (SMA)	Simple Moving Average of Adjusted Close price over a period, serving as a baseline for Bollinger Bands.
Upper Band	Middle Band + (k × standard deviation), highlighting overbought market conditions.

Indicator	Description
Lower Band	Middle Band − (k × standard deviation), highlighting oversold market conditions.
Fibonacci Levels	Derived from high and low prices, marking potential retracement and support/resistance zones.
Ichimoku Cloud	Comprehensive indicator providing trend direction, support/resistance, and momentum analysis.
CCI (Commodity Channel Index)	Measures deviation of current price from historical average to identify overbought/oversold conditions.
RSI (Relative Strength Index)	Oscillator tracking price momentum to identify overbought (>70) or oversold (<30) conditions.

Target Variables

Next day's price as the **target** can lead to the **naive persistence problem.**

This may result in low error metrics but fails to capture meaningful trading patterns.

Variable	Description
Target	The difference between the **Adjusted Close** price and the next window-size **Adjusted Close** price.
Pct Return	The total percentage return over the time window, calculated using the **Adjusted Close** price.
Target Class	Classification label based on **percentage return distribution**: **1 (Up)** if > 66th percentile; **0 (Down)** if < 33rd percentile; **2 (Neutral)** otherwise.

Proposed solution:

- **Classification**
- **(Up, Down, Neutral)** based on percentage return distribution

Data Preprocessing & EDA

Data Cleaning

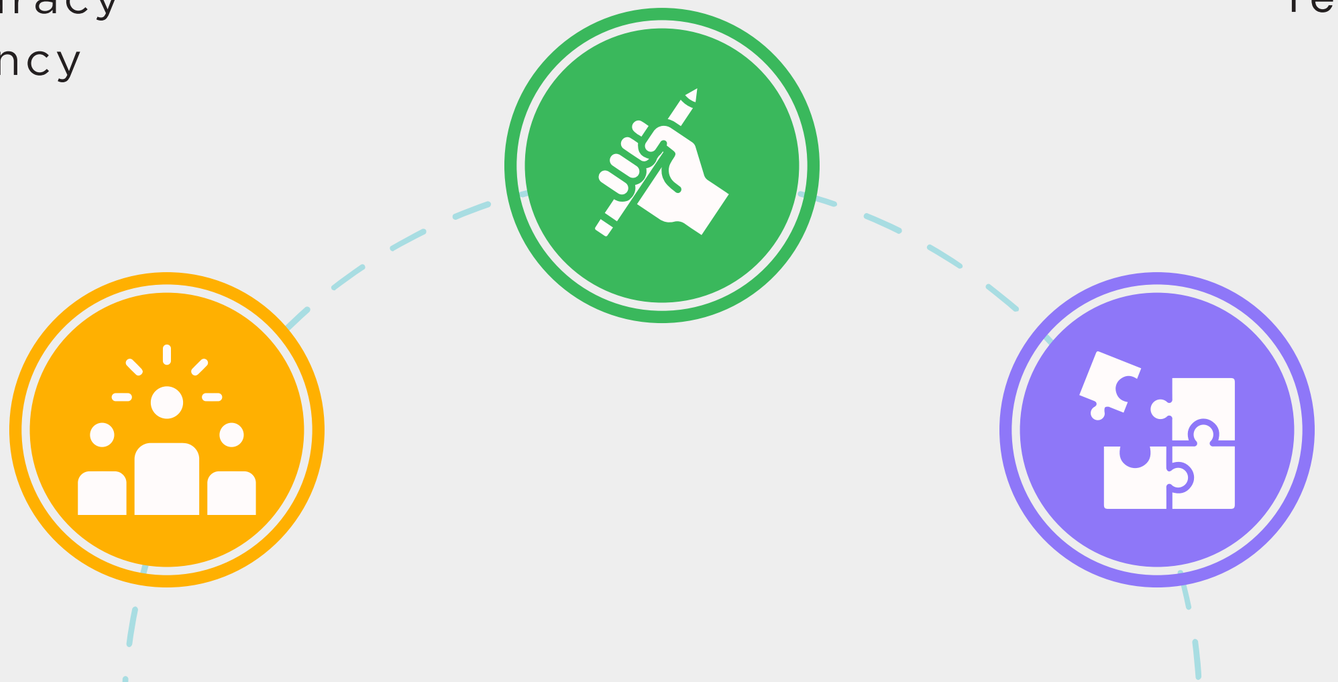
Removing incorrect or incomplete data to ensure accuracy and consistency

Data Transformation

Encoding variables and doing feature engineering

Exploratory Analysis

Analyze data to discover patterns and relationships



Data Cleaning

Initial Fix:

- Removed the **first two rows**, which were improperly imported as a mix of cryptocurrency names and column headers.

Column Renaming:

- Standardized column names for each cryptocurrency to ensure clarity and uniformity.

Consistency Check:

- Verified the dataset for matching dimensions, consistent
- Count of NAs (no missing values)

Missing Values:

Cryptocurrency	Open	High	Low	Adj. Close	Volume
ETH-USD	0	0	0	0	0
BTC-USD	0	0	0	0	0
USDT-USD	0	0	0	0	0
DOGE-USD	0	0	0	0	0

Dimensionality Check

Cryptocurrency	Dimensions
ETH-USD	(2557, 5)
BTC-USD	(2557, 5)
USDT-USD	(2557, 5)
DOGE-USD	(2557, 5)

Data Transformation

Target Variables:

- Encoded key target variables
  - Price Difference
  - Percentage Return (Pct Return)
  - Target Class (Up, Down, Neutral)

Technical Indicators:

- **Added 22 features**, incorporating advanced technical indicators (e.g., EMA, RSI, Bollinger Bands)***.***

Missing Values:

- Caused by technical indicator calculations.
- Removed rows with missing values
- Resulting in 2382 observations,

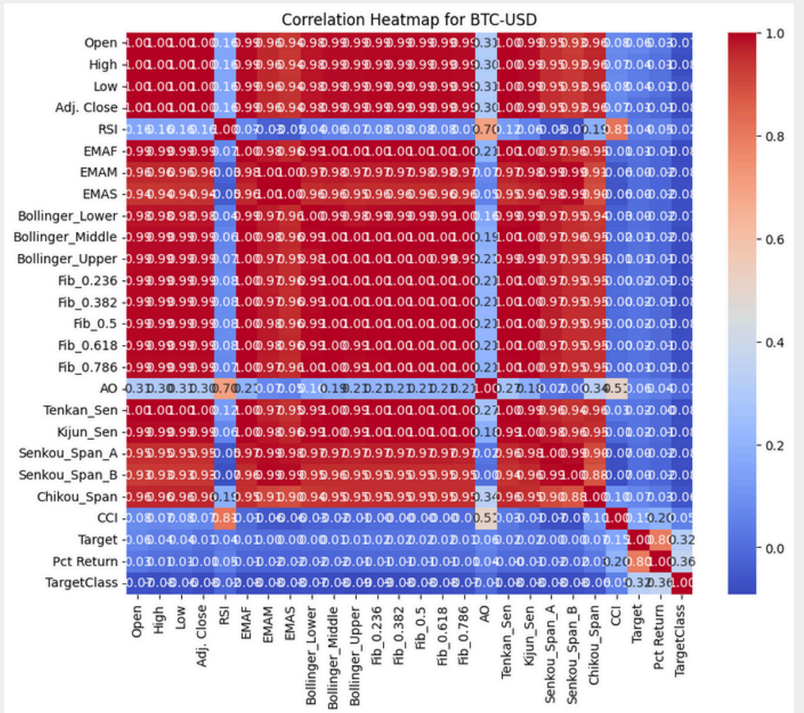
Metric	Value
Total Observations	2382
Time Span	~6 years
Features Included	26
Data Removed (NA)	~0.06%

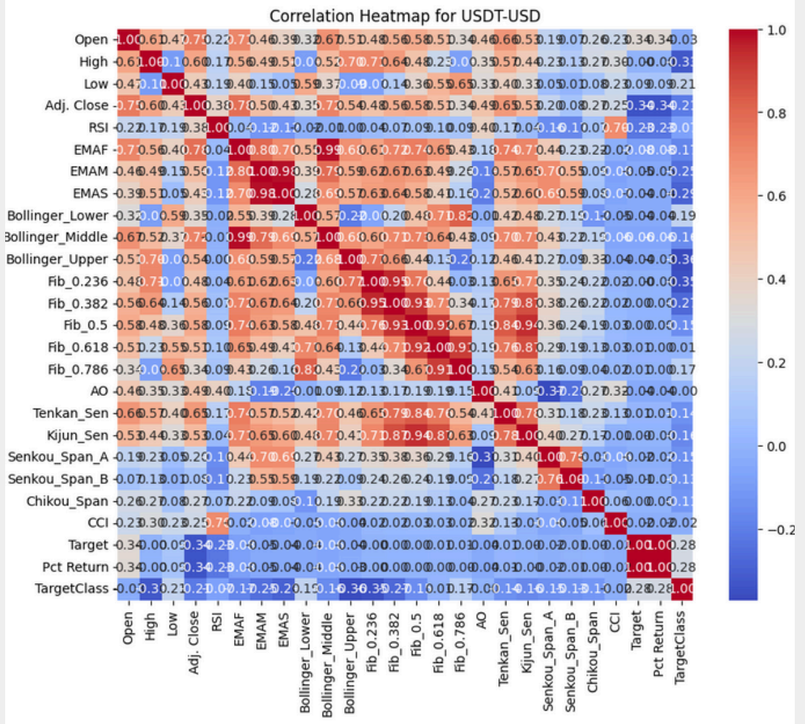
Exploratory Data Analysis

Correlation Analysis:

- **BTC - DOGE - ETH**
  - **Highly Correlated Features:** Open, High, Low, Adj. Close, EMAF, EMAM, EMAS—interconnected by nature.
  - **Target Variables:** Low correlation with Target and Pct Return—captures future price movement, not current data.
  - **RSI:** Low correlation—adds unique momentum insights, valuable for the model.

- **USDT:**
  - **Moderate Correlation:** Price features (High, Low, Adj. Close) reflect USDT's stability.
  - **Indicators:** Less effective due to low volatility and predictable pricing.





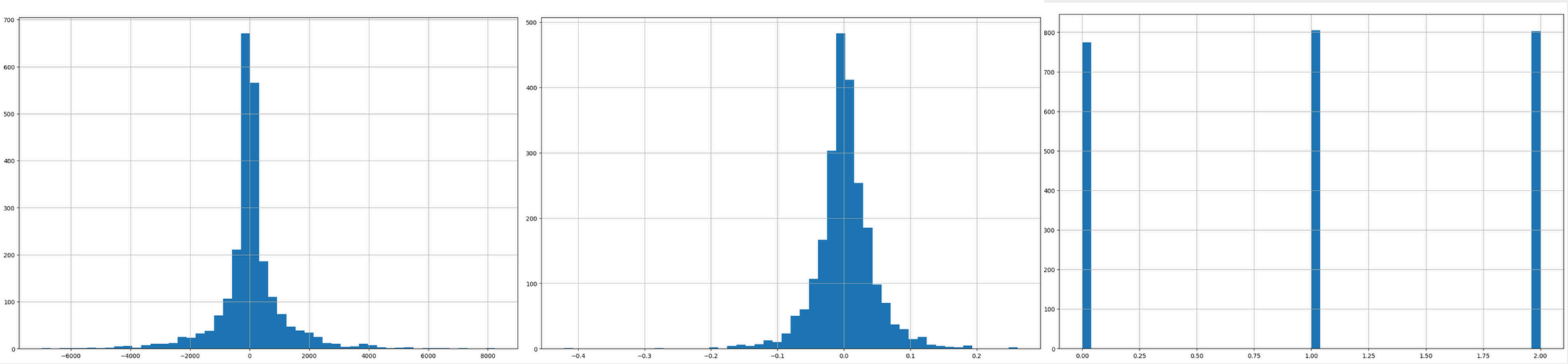
Exploratory Data Analysis (2)

Price Differences/ Pct Return distribution:

- **Centered at zero:** positive and negative returns balance out over time.
- No strong upward or downward trend

Target Class:

- **Balanced**
- It allows for better metrics computation



Feature Selection

- Feature selection is crucial to **reduce dimensionality**, eliminate r**edundant features,** and enhance **model generalization**.
- **Random Forest was chosen** as it captures non-linear relationships, retains time-series dependencies, and provides reliable feature importance using Gini Importance.
- The process includes **training the model**, evaluating **feature importance**, removing **low-importance features.**
- The features that had **importance scores lower than the 25th percentile** were removed, keeping only the more impactful ones.
- The number of features was slightly reduced

```
# Define model; Baseline model: no hyperparamether tuning;
model = RandomForestClassifier(n_estimators=100, random_state=42)
```

Feature	ETH-USD	BTC-USD	USDT-USD	DOGE-USD
Open	0.080238	0.078144	0.131787	0.080491
High	0.050646	0.051431	0.071144	0.056475
Low	0.052713	0.050588	0.054464	0.049507
Adj. Close	0.064399	0.067699	0.107315	0.068770
RSI	0.071317	0.071534	0.082029	0.068888
EMAF	0.041274	0.040327	0.040719	0.040060
EMAM	0.041818	0.042804	0.033905	0.040863
EMAS	0.041930	0.044803	0.039623	0.041786
Bollinger_Lower	0.049057	0.047237	0.034503	0.044649
Bollinger_Middle	0.039231	0.038080	0.030616	0.039355
Bollinger_Upper	0.044037	0.042965	0.074046	0.045978
AO	0.053233	0.056231	0.033882	0.050101
Kijun_Sen	N/A	N/A	N/A	0.028846
Senkou_Span_A	0.041727	0.041353	N/A	0.041066
Senkou_Span_B	0.029245	0.028901	N/A	N/A
Fib_0.236	N/A	N/A	0.036926	N/A
Fib_0.382	N/A	N/A	0.026416	N/A
Fib_0.786	N/A	N/A	0.025411	N/A
Chikou_Span	0.055765	0.055780	0.029176	0.053772
CCI	0.068410	0.069922	0.033805	0.068089

Multiple Linear Regression

- Regression-based approach to predict **daily price differences** as a continuous variable.
- This results serve as a baseline
- They highligh the complexity and volatility of price movements, justifying the **need for classification-based** models.

- Evident **overfitting** and the inability to generalize due to noise.
- Addressed multicollinearity: **removed highly correlated features**
- Maintained **time order** for realistic predictions
- Evaluated performance with **RMSE and R^2**, providing insights into model accuracy and limitations.

Metric	ETH-USD	BTC-USD	DOGE-USD	USDT-USD
Train RMSE	64.4321	818.2320	0.0096	0.0027
Test RMSE	75.5992	1334.0173	0.0091	0.0004
Train R^2	0.3332	0.3302	0.3679	0.3346
Test R^2	0.2882	0.2835	-0.1401	-0.2530
Intercept	86.1895	1071.1940	0.009971	0.2130
Coefficients	[-0.3895, -1.7622, ...]	[-0.3551, -21.8469, ...]	[-0.4906, -0.00019, ...]	[-0.6610, -0.00012, ...]

```
# Train-test split (80% train, 20% test, keeping time order)
split_idx = int(len(df) * 0.8)
X_train, X_test = X[:split_idx], X[split_idx:]
y_train, y_test = y[:split_idx], y[split_idx:]
```

```
# Train the Linear Regression model
model = LinearRegression()
model.fit(X_train, y_train)
```

Why LSTM

- LSTM is a modified version of RNN, that is the best model to evaluate time-series patterns, that overcome the problem of reducing importance of weights over time.

- **Lookback window** we will use 50 in order to capture both long and short term patterns

- Data were **shaped** in order to **match feature selections** and **LSTM**() expected format , on both training and test set.

```
Dataset: ETH-USD
(1865, 50, 17)
(1865,)
(467, 50, 17)
(467,)
Dataset: BTC-USD
(1865, 50, 17)
(1865,)
(467, 50, 17)
(467,)
Dataset: USDT-USD
(1865, 50, 17)
(1865,)
(467, 50, 17)
(467,)
Dataset: DOGE-USD
(1865, 50, 17)
(1865,)
(467, 50, 17)
(467,)
```

Model and Optimization

Hyperparameter

- Number of layers = (1, 3)
- Units per layers = (32, 64, 128)
- DropoutRate = from 0.1 to 0.5
- Learning Rate = (0.001, 0.005, 0.0001)

- We use **Bayesan Optimizer** with val_accuracy as parameter to maximize, the model training is performed using early stopping in order to prevent overfit

```
Best val_accuracy So Far: 0.4477211833000183
```

- Overfitting
- The low validation accuracy across most of the trial suggest that the model is not learning underlying patterns
- The Loss value for both training and validation are high

Classification Rules

- Even **small changes** to the classification rules might result in **wildly different outputs.**
- They act as a **link** between a model **continuos probability** and its final, **discrete direction (labels).**

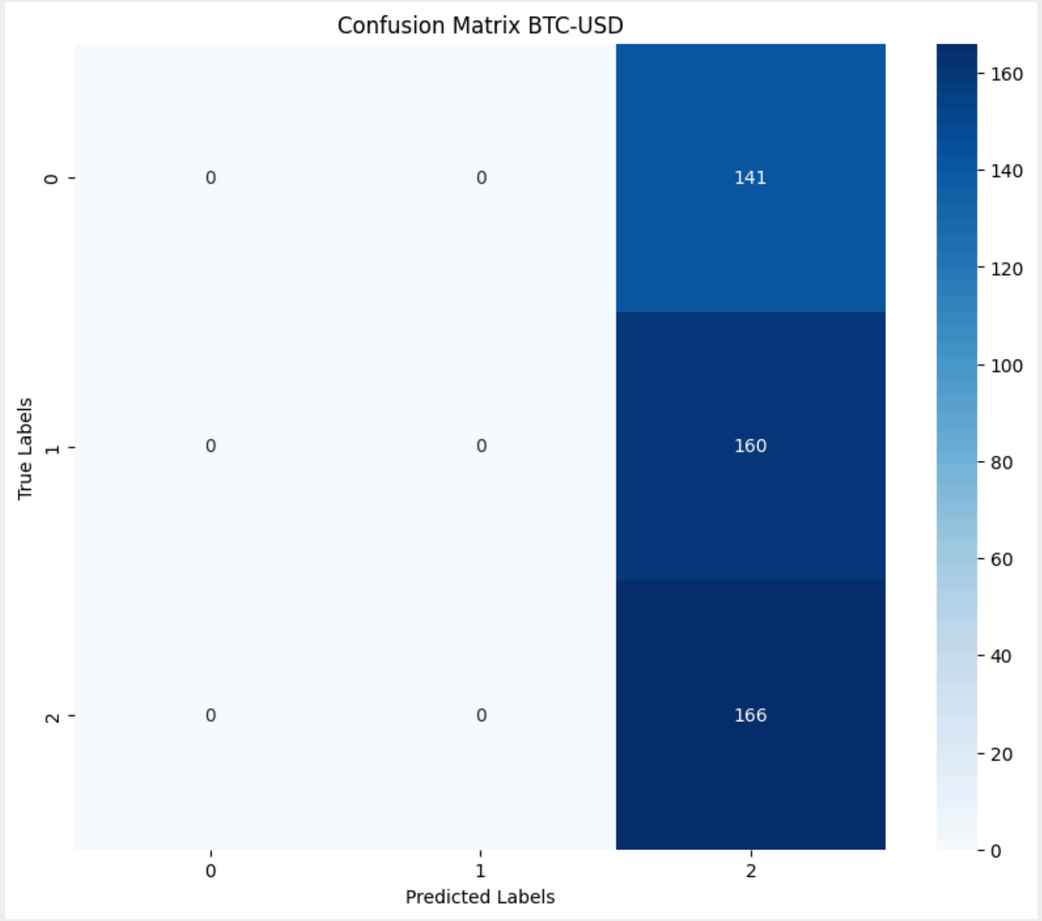
We utilized and compared results for **4 different classification rule:**

- *ArgMax*
- *Percentile 33th and 66th*
- *Percentile 30th and 70th*
- Percentile 33th and 66th + Buffering

Classification Rule (1)

- The Argmax interprets the model's output as its confidence in each class.
- It assume that the highest probability represent the actual underlying pattern

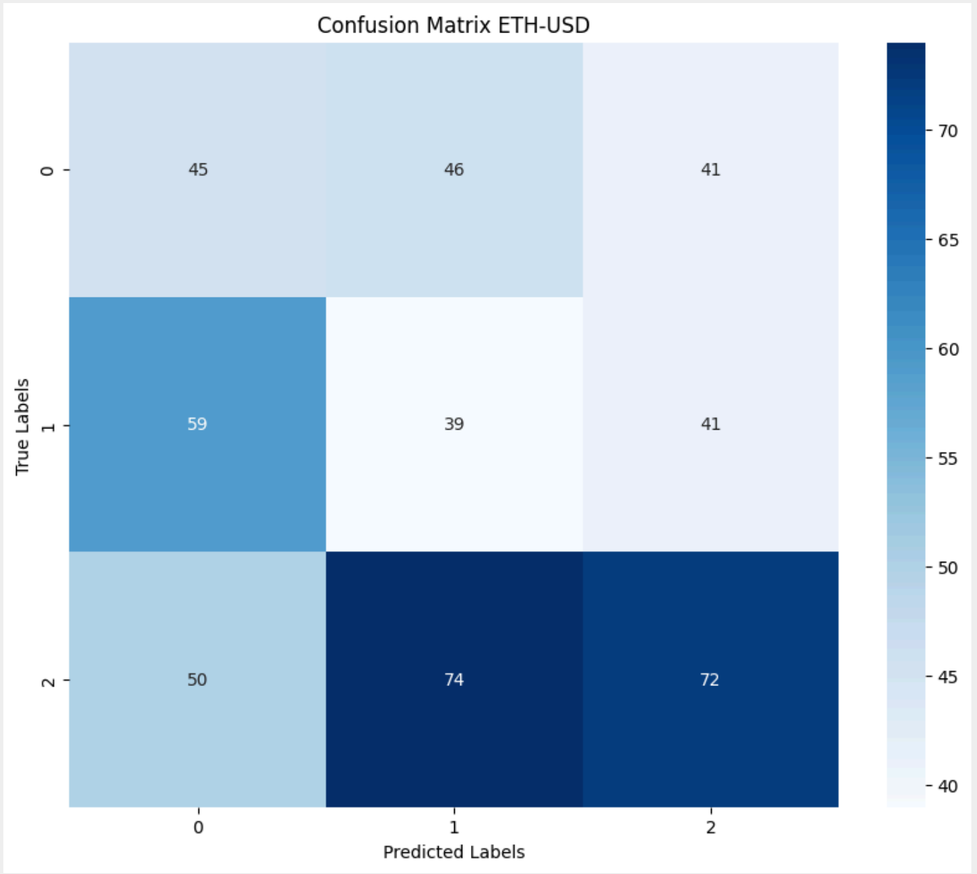
- **Limitation:** This method **cannot take uncertainty** into account, if the probability are balanced there is a very high missclassification



Classification Rule (2)

- Evaluate **upper** and **down threshold** based on the distribution of the maximum probability of each observation
- Up-thresh = 66th percentile
- Down-thresh = 33th percentile

- **Limitation**: If distribution is skewed an sample size is small, fixed percentile may not adjust wee to changes in probability distribution

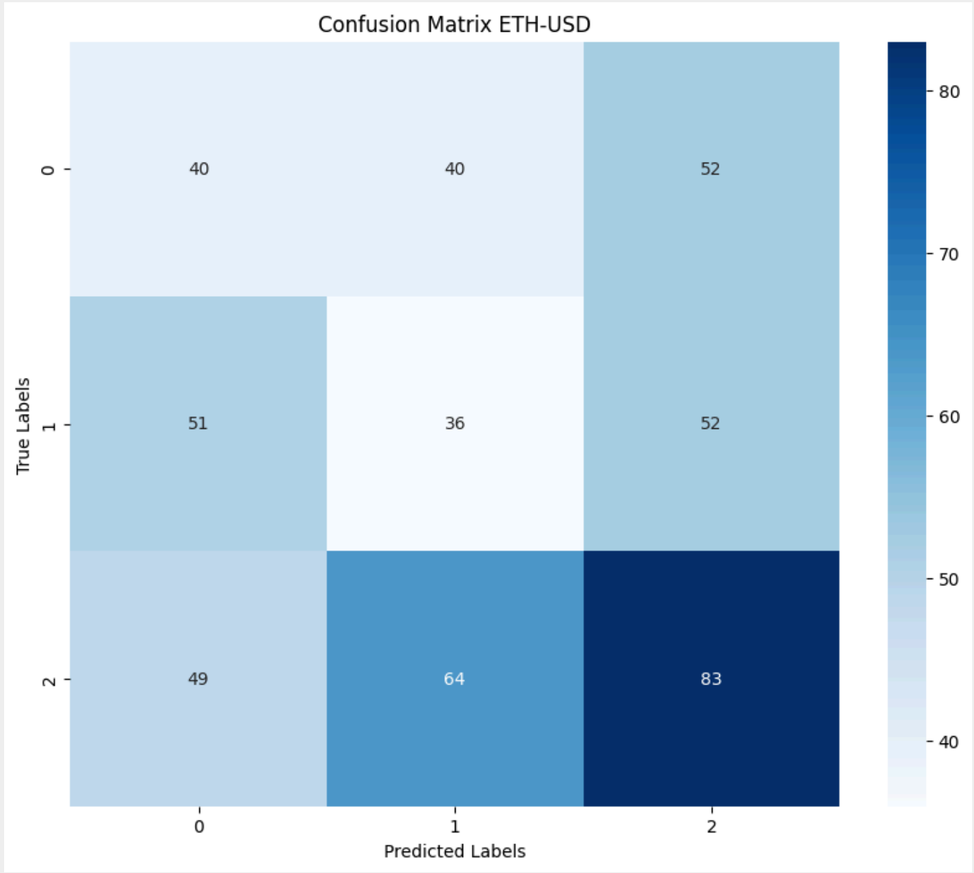


Classification Rule (3)

- Interprets output probability giving a more distinction between upper and lower threshold

- Up-thresh = 70th percentile
- Down-thresh = 30th percentile

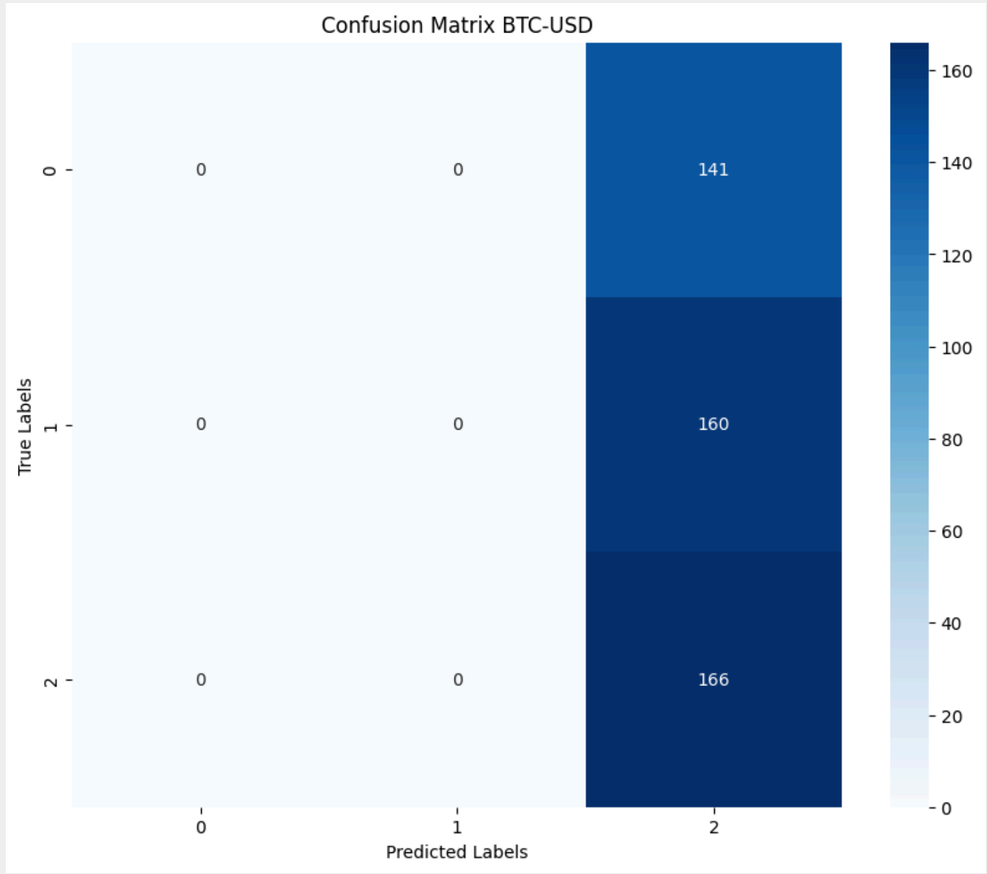
- The goal is to enhance classification balance by expanding the middle category



Classification Rule (4)

- Buffered Percentile Rule, extension of the percentile, is used in cases when the probability are extremely near.
- By adding another layer of caution to the probability's interpretation

- Limitation: It needs careful calibration, an incorrect adjusted buffer might add bias or reduce flexibility



Key Findings:

Class imbalance remained a major issue across all methods, models struggled to predict class(0) down and class(1) up and often favoring neutral (2)

The **Argmax** classifier wast the worst performing reducing classification flexibility and create rigid and biased predictions

The **best** performing classification was the one that utilize 70th and 30th threshold providing the most balanced results

Random Forest Classifier

- **Data Scaling:** All features were scaled prior to model training.
- **Hyperparameter Tuning**: **Bayesian optimization** was employed to identify the best parameters for the models.
- **Classifier Variants:**
  - **Model 1:** Implemented Random Forest using the best parameters from the optimizer **(No further feature selection).**
  - **Model 2:** If overfitting was detected, **further feature selection** was performed using the median importance threshold. The optimizer was rerun, and a new Random Forest model was trained on the updated features and parameters.
  - **Model 3 (Variant of Model 2)**: If overfitting detected, feature selection was repeated with a **fixed importance threshold of 5%**. The optimizer and model training processes were then repeated.

Model 1:

Dataset	Train Accuracy	Test Accuracy
ETH-USD	0.8530	0.4277
BTC-USD	0.8541	0.4654
USDT-USD	0.9407	0.7862
DOGE-USD	0.8940	0.5073

Model 2:

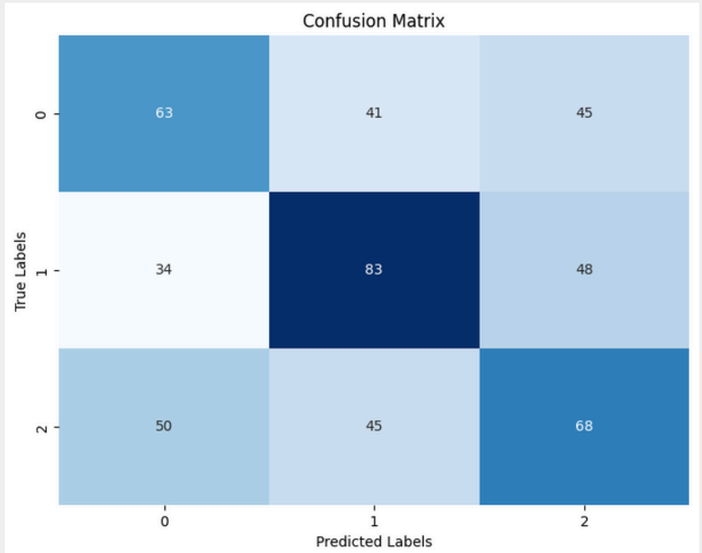
Dataset	Train Accuracy	Test Accuracy
ETH-USD	0.9186	0.5828
BTC-USD	0.8940	0.5430
USDT-USD	0.9349	0.8071
DOGE-USD	0.8751	0.5241

Model 3:

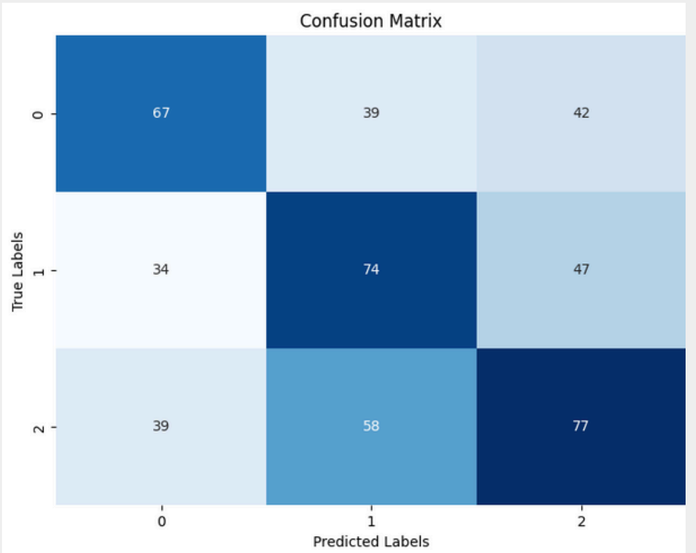
Dataset	Train Accuracy	Test Accuracy
ETH-USD	0.8551	0.4591
BTC-USD	0.8903	0.4801
USDT-USD	0.9323	0.8155
DOGE-USD	0.9076	0.5744

Model 1

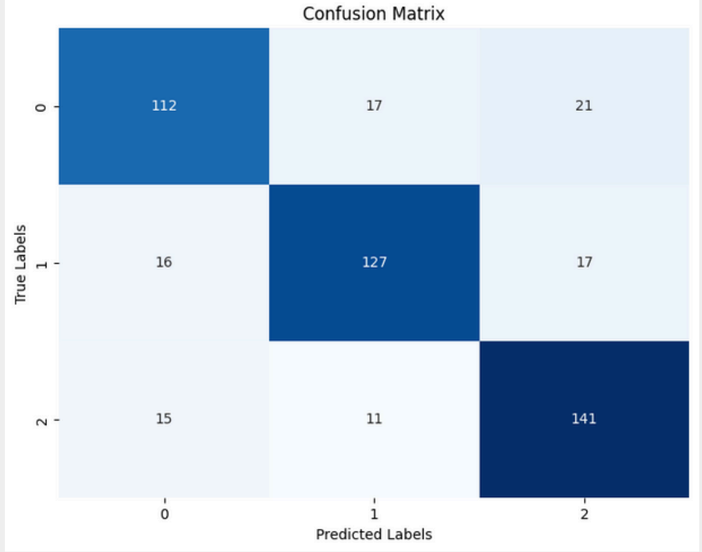
ETH



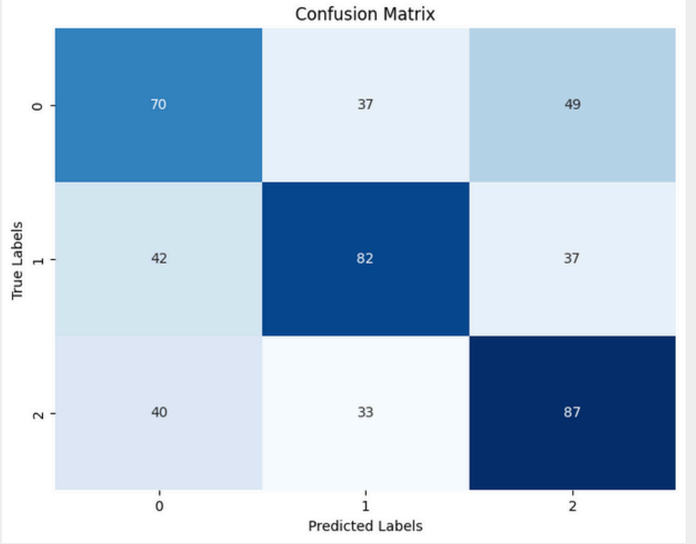
BTC



USDT

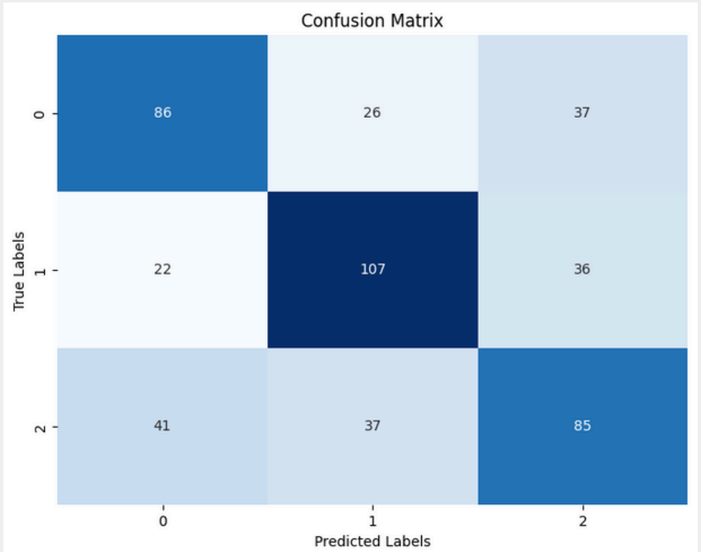


DOGE

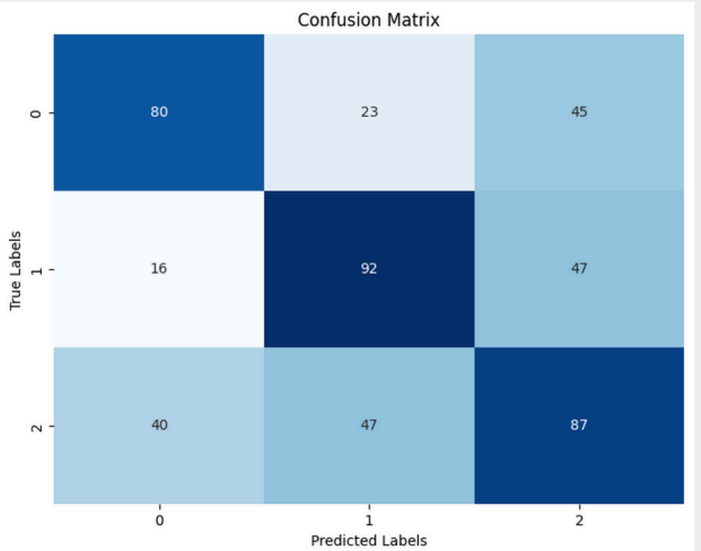


Model 2

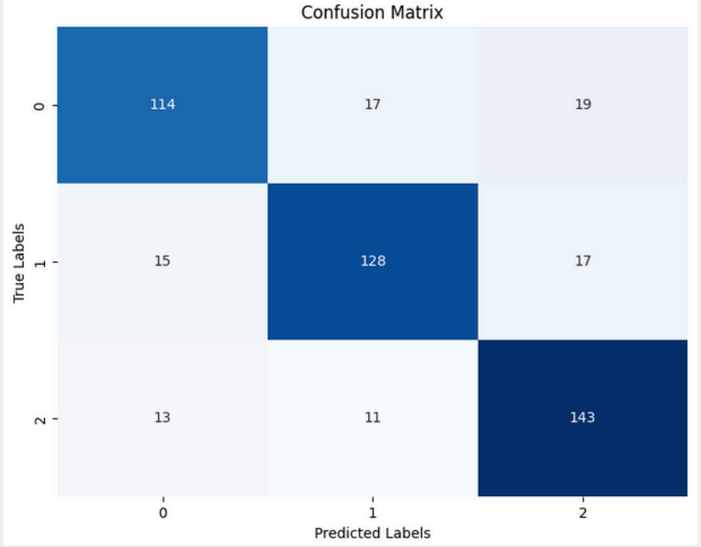
ETH



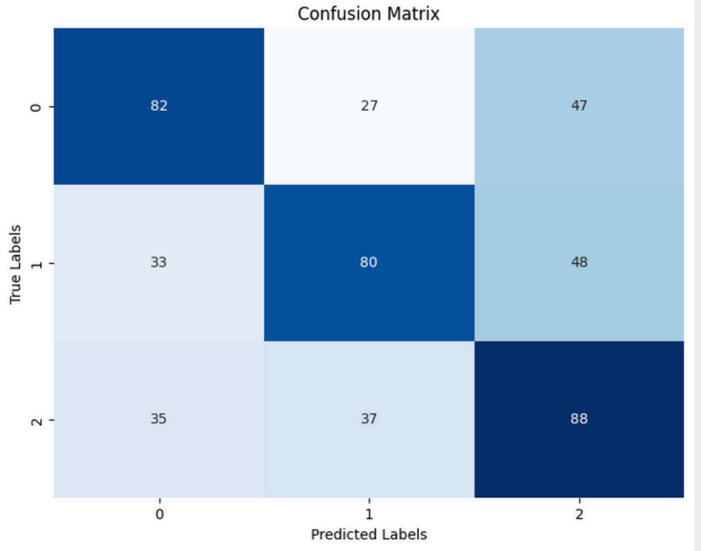
BTC



USDT



DOGE



Portfolio Optimization

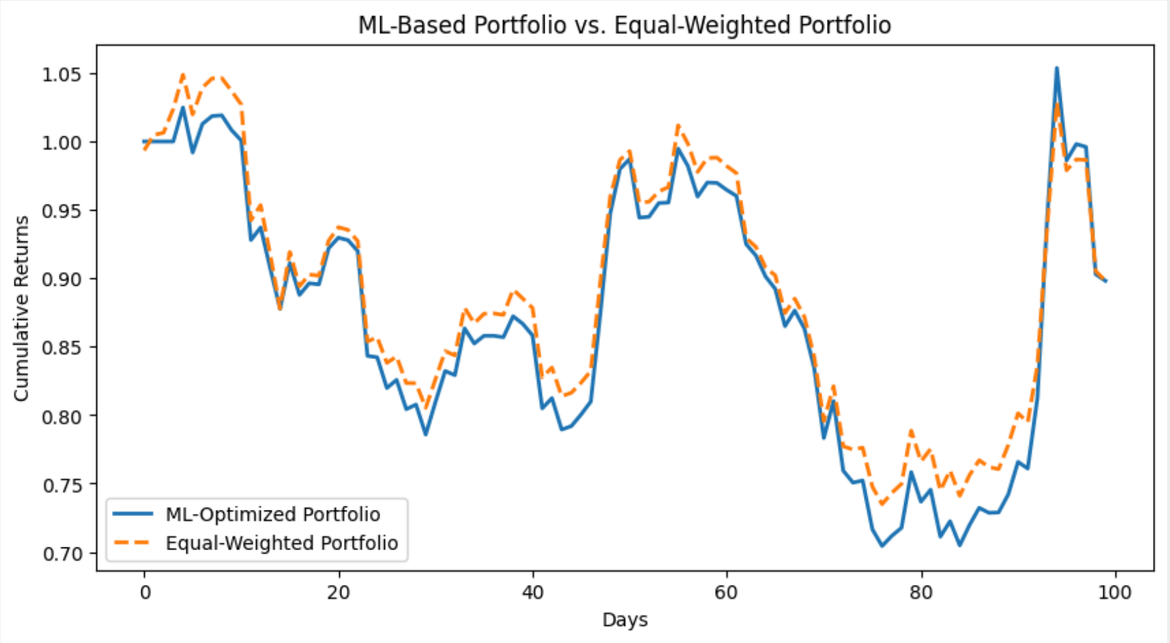
Investment allocation is dynamically adjusted based on RandomForestClassifier predictions for asset movements (up, down, neutral).

Investment Strategy:
Up → 50% allocation
Neutral → 30% allocation
Down → 20% allocation

Weights are normalized to prevent exceeding 100%.

Portfolio Growth Calculation:

- Returns weighted by historical percentage movements.
- Summed weighted returns define total portfolio performance.
- Cumulative return computed to track overall portfolio growth.



Since the model predicts movement direction but not magnitude, traditional optimization methods were unsuitable. Instead, this approach prioritizes capital allocation based on probability of positive movement

Key findings & Limitations

Challenges - Limitations:

- **Model Overfitting**: Need to refine feature selection & regularization.
- **Market Volatility**: Crypto markets can be unpredictable, limiting model reliability.
- **External Factors**: News, sentiment, and macroeconomic events impact prices beyond historical data.

Key findings:

- ML-based portfolio allocation is a promising strategy but needs refinement.
- Achieved **competitive performance vs. equal-weighted portfolios**, showing potential in volatile markets.
- Future work should focus on **enhancing prediction accuracy** and **risk management** to maximize returns.

Thank you for the attention.

ANTONELLA FACINI 900455
GIORGIO BERNACCHI MONTI 886263