

# Крипто прогнозиране & изкуствен интелект

„Прогнозирането на финансовите пазари е като да караш кола със закрито предно стъкло, гледайки само в огледалото за задно виждане.“

Изготвили:

Велизар Митов

Десислава Недялкова

Иван Паспалджиев

Иван Резняков

Камелия Стефанова

Николай Пенчев



# План на работа

- ▶ Архитектура на проекта
- ▶ Използвани модели за предсказване
- ▶ Описателна статистика
- ▶ Feature engineering ARIMA & VAR via Orange
- ▶ Използвани модели (ML: LSTM и хибриден)
- ▶ Визуализация (Plotly, Matplotlib)
- ▶ Оптимизация на портфейл с генетичен алгоритъм
- ▶ Backtest (модул метрики (MAPE, DS, R, G))
- ▶ Финален код review



# Описателна статистика

[illegible]



Filter...

SELECT

INTERPRET

LAUNCH APP

UNPLAY

Context

Count

Length

Variety

Cooccurrence

Category

Convert

Message

Interchange

Time Series

Yahoo Finance

Form Timeseries

Interpolate

Moving Transform

Line Chart

Periodogram

Correlogram

Spiralogram

Granger Causality

ARIMA Model

VAR Model

Model Evaluation

Time Slice

Difference

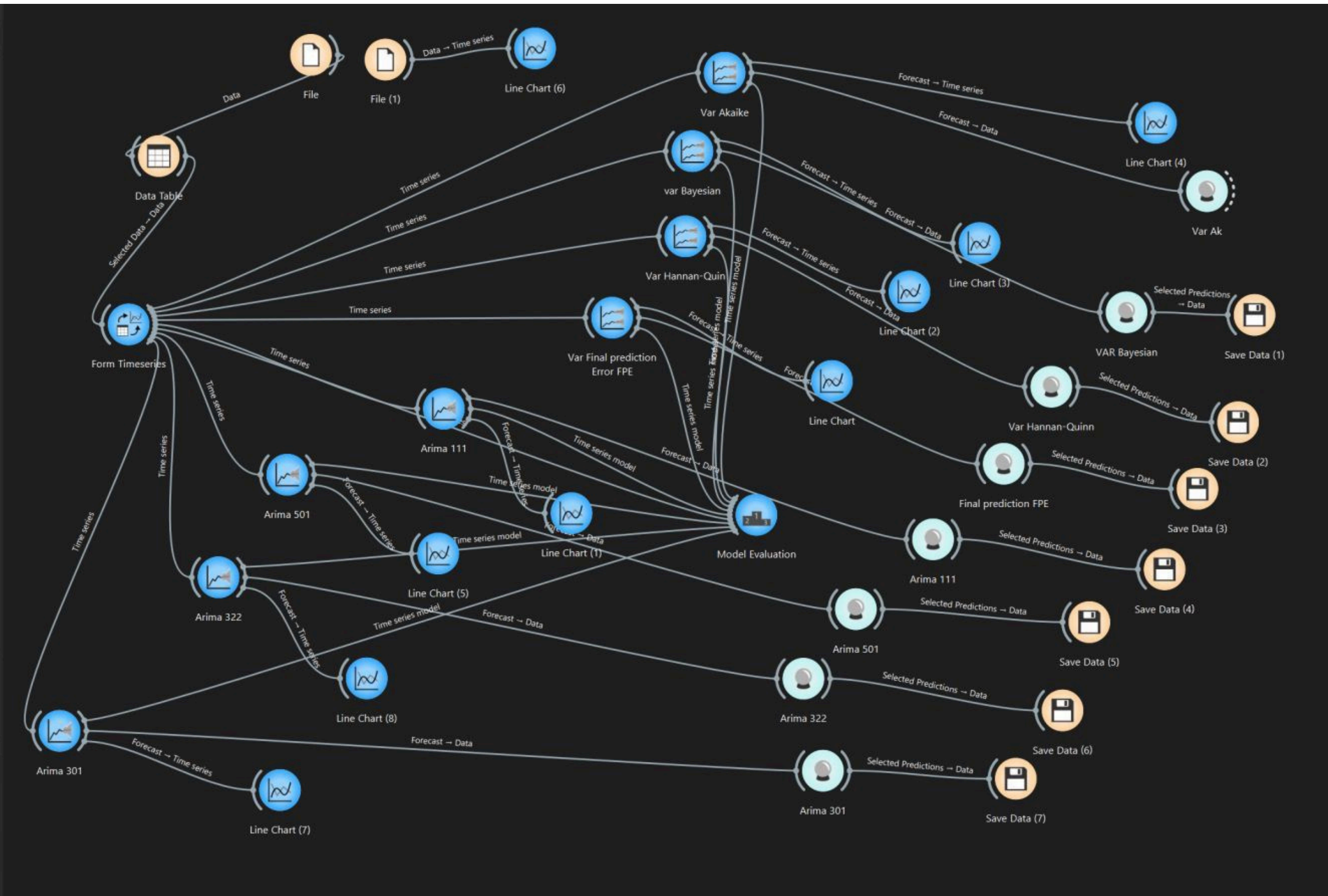
Seasonal Adjustment

World Happiness

Save Data

Save data to an output file.

more...



Adrian Loeb  
CEO & Founder

# ВЪВЕДЕНИЕ

## Архитектура на проекта в контекста на CRISP-DM

Cross Industry Standard Process for Data Mining  
(Стандартизиран междусекторен процес за извличане на знания от данни)  
CRISP-DM е често използван в проекти с данни, включително и такива за предсказване на цени на криптовалути.

☛ Целта ни е да изградим модел за прогноза на BTC цената на биткойни и Rule-based AI трейдър за BTC (начално ниво).

☛ Проблемът е как да прогнозираме BTC цената една стъпка напред и да вземаме базови автоматизирани инвестиционни решения

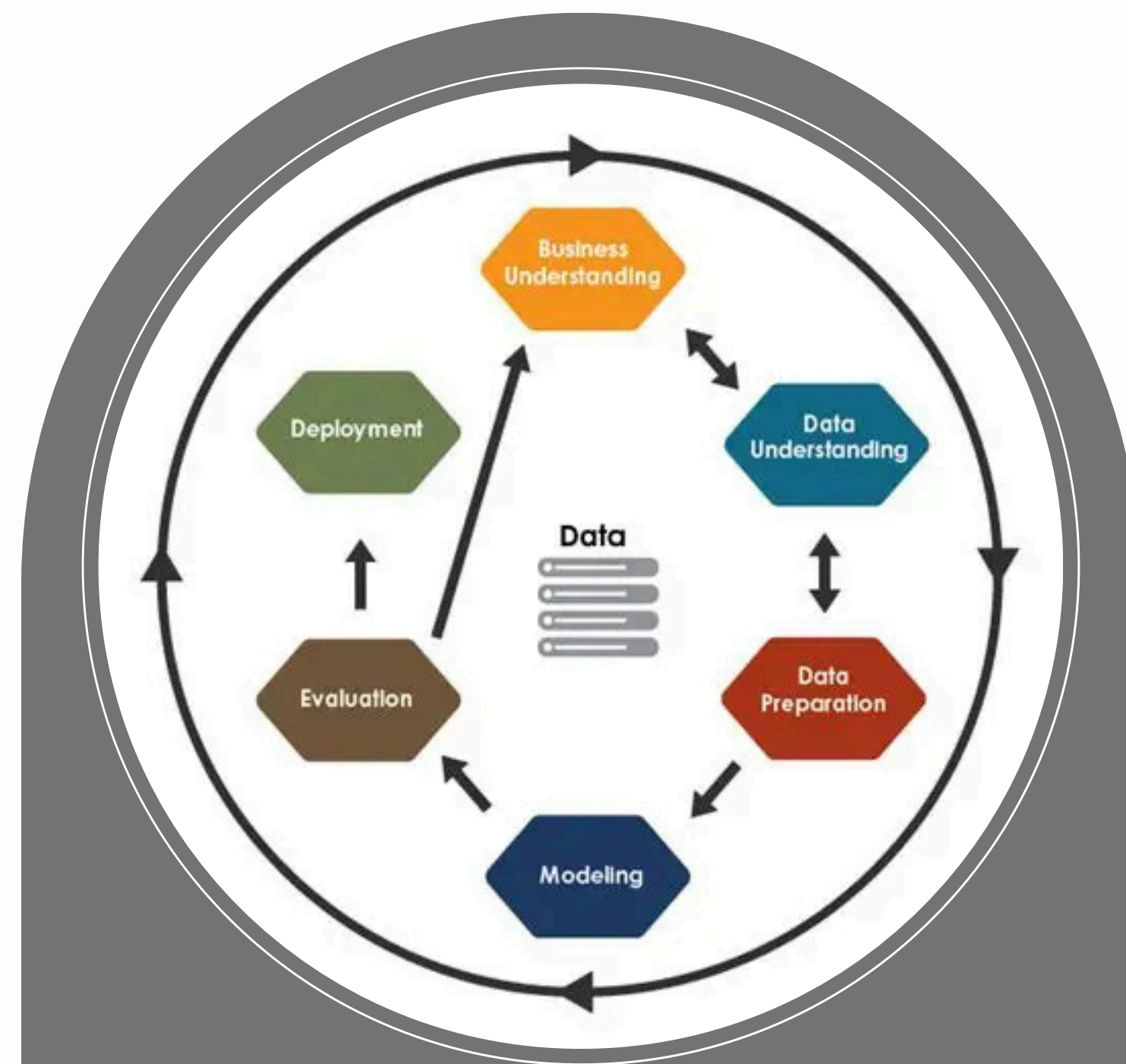
☛ Разбиране на данните. Исторически данни на последните 10 години от Investing.com

☛ Подготовка на данните. Описателна статистика и хисторграма

☛ Моделиране. Моделите са ARIMA via Orange & VAR via Orange

☛ Оценка. Валидация на база различни статистики.

☛ Разгръщане. Model training



# LSTM (Long Short-Term Memory) (Дългосрочна краткосрочна памет) е тип рекурентна невронна мрежа (RNN) recurrent neural network

Архитектурата на изкуствена невронна мрежа работи с последователности от данни, като например времеви редове (time series) за

- Предвиждане на стойности във времето (например цени на биткойн)
- Анализ на последователни данни (напр. движения на пазара, реч, текст)

Работи опростено както следва

## Входна врата

решава каква нова информация да влезе в клетката.

## Забравяща врата

решава коя информация от предишното състояние да "забрави".

## Изходна врата

определя каква информация да се изпрати като изход.

Защо LSTM е подходящ за криптовалути?

- Моделът може да запомня пазарни движения и модели във времето.
- Устойчив е на шум в данните, характерен за крипторынокът.
- Подходящ е за краткосрочни и дългосрочни предсказания.



# Използван модел LSTM

Моделът, който използваме е за да направим ценово прогнозиране за пазара на биокойни. Ние експериментираме с хибриден модел за дълбочинно обучение, който комбинира CNN, части от трансформатори и LSTM. Нашата цел е да направим прогноза за следващата цена, използвайки само минали данни и текущо налични данни, за да избегнем изтичане на информация от бъдещето.

```
#Causal Conv1D: Avoids future data, good for fast signal extraction.
#Multi-Head Attention: Transformer-style attention to focus on important parts of the sequence (no future leak).
#LSTM: Captures temporal dependencies.
#Dense: Outputs 1 value (predicted delta).
#Loss is MSE, but the model uses sample_weight=confidence, giving more influence to high-confidence examples.
def build_causal_price_predictor(input_shape):
    from tensorflow.keras.models import Model
    from tensorflow.keras.layers import Input, Dense, Conv1D, Add, LayerNormalization, MultiHeadAttention, LSTM, Dropout

    inp = Input(shape=input_shape)

    # Causal convolution: looks only at current and past
    x = Conv1D(32, kernel_size=1, padding='causal', activation='relu')(inp)

    # Transformer-style attention (no future access)
    attn = MultiHeadAttention(num_heads=2, key_dim=32)(x, x, attention_mask=None)
    x = Add()([x, attn])
    x = LayerNormalization()(x)
    x = Dropout(0.3)(x)

    # Single-direction LSTM (causal)
    x = LSTM(64, return_sequences=False)(x)
    x = Dropout(0.3)(x)

    # Final linear regression output
    out = Dense(1, activation='linear')(x)

    model = Model(inputs=inp, outputs=out)
    model.compile(optimizer='adam', loss='mse', metrics=['mae'])
    return model
```

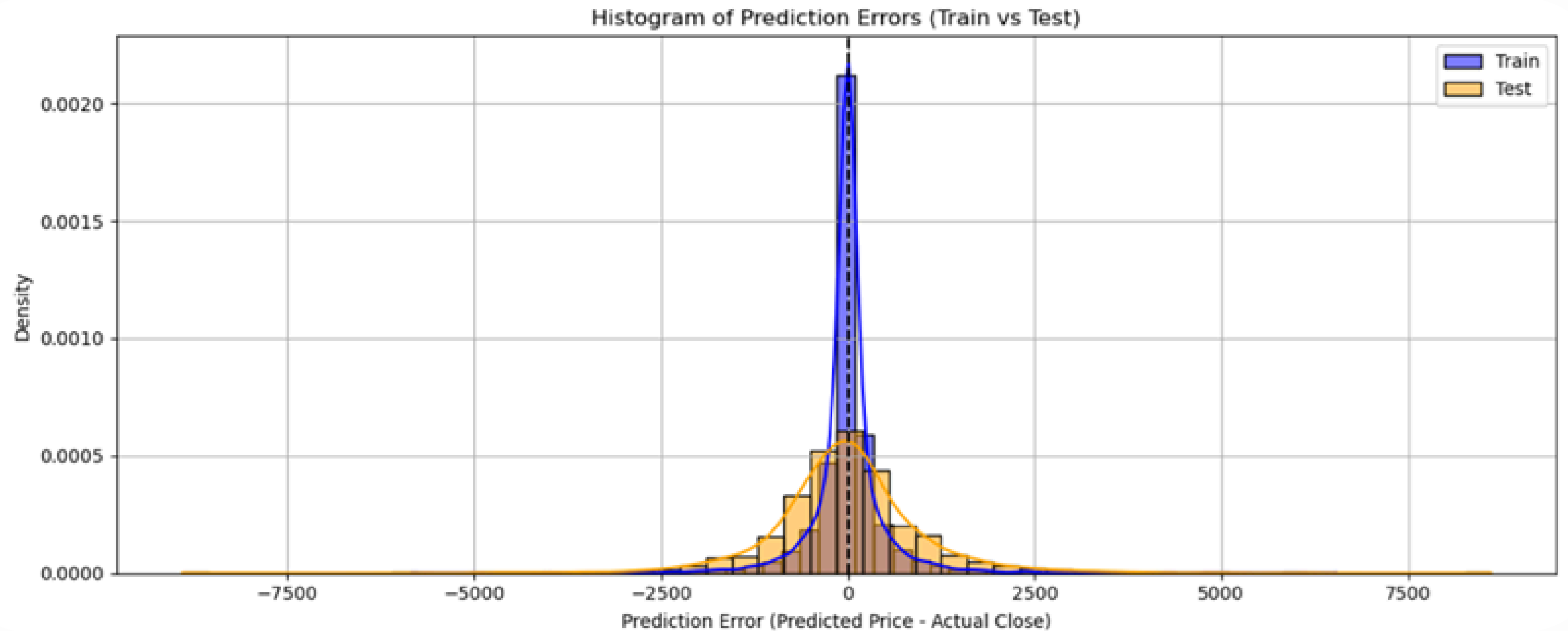
## Прогнозирана срещу реална цена на затваряне



От картината можем да видим, че предсказаните от нас цени почти перфектно покриват пазарните стойности с малки отклонения, но за да може стратегията да работи, ще ни е нужна допълнителна логика за предсказването на цените, за да може моделът да бъде печеливш, тъй като частта ни за класификация на посоката не е достатъчно добра.

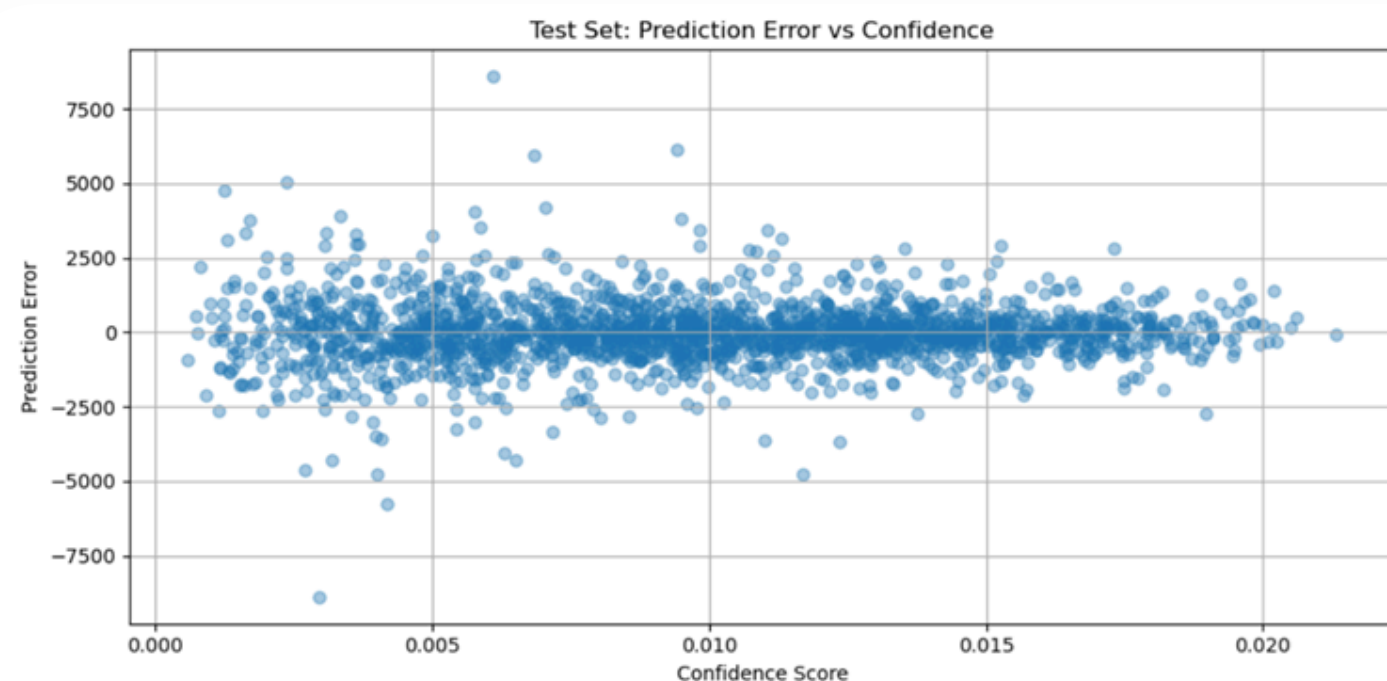


## Хистограма на грешките в предсказанията

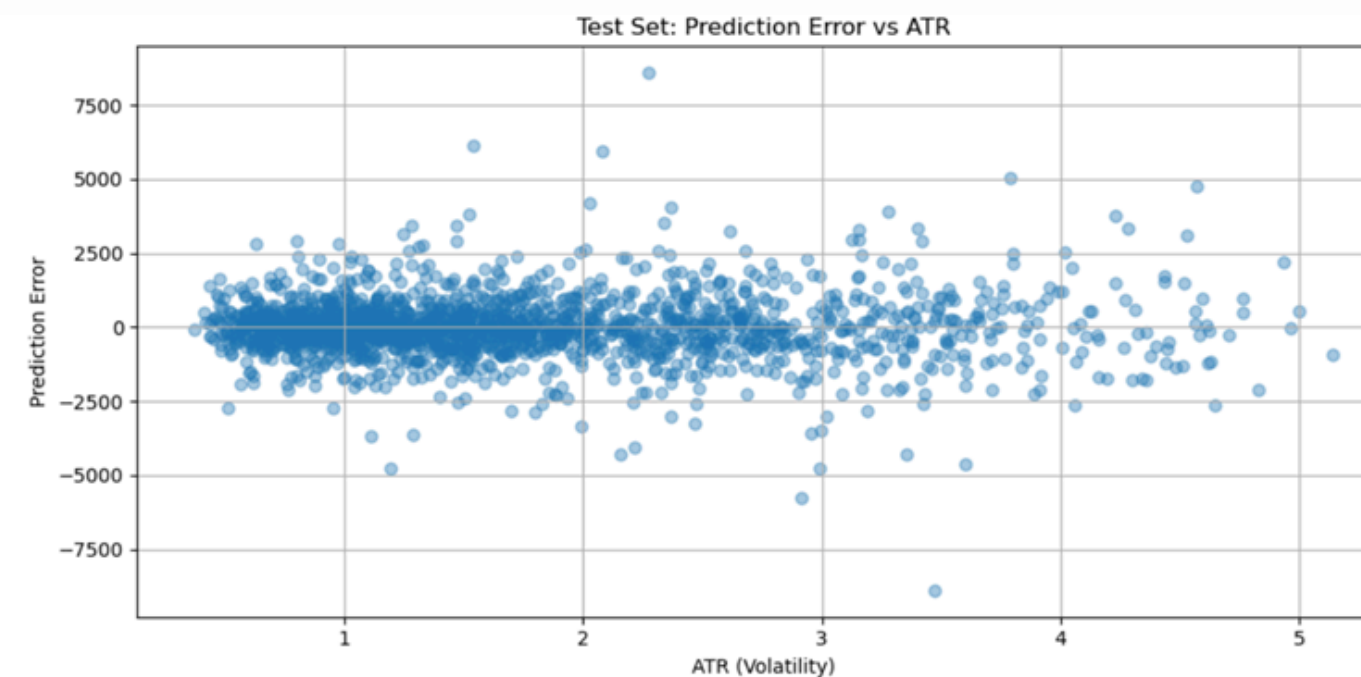


Таблица, описваща грешките в прогнозите за обучаващия и тестовия набор. От тази графика можем да видим, че моделът работи добре.

# Тестова прогнозна грешка vs увереност



Графиката е диагностичен инструмент за валидиране на това дали оценките на увереността корелират с надеждността на прогнозите. Виждаме, че с нарастване на увереността, грешките намаляват. Виждаме, че при широко разпространение на грешките при ниска увереност, стесняващо се до нула грешки при висока увереност. долния график. Анализирайки как волатилността (ATR) влияе на прогнозната грешка на нашия модел. Целта е да се види как пазарната волатилност влияе на надеждността на модела.

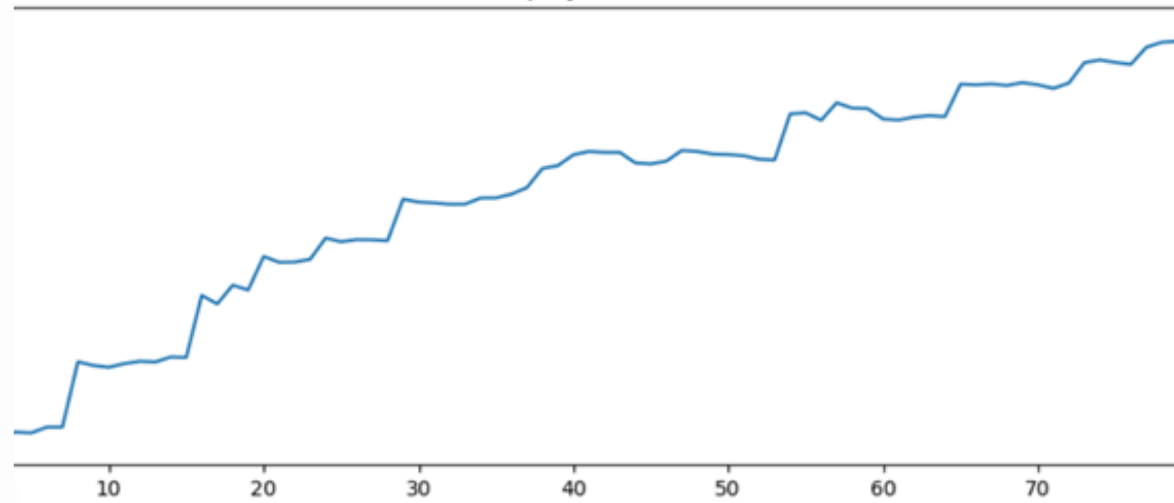


Най-добрият резултат би бил формата на фуния с тесни грешки при нисък ATR, без ясна възходяща тенденция в високите региони моделът да не експлодира с волатилност. следващото нещо, което правим, е да стартираме персонализирана търговска симулация. Ние определяме нашата посока на базата на промяна в цената.

# Първи опит за предсказания на пазара (Back Testing)

```
.27.732702631352, 'total_pnl': 227.73270263135237, 'mean_pnl_per_trade': 2.8115148473006126, 'sharpe_ratio':  
.62124823363, 'win_rate': 0.5432098765432098, 'max_win_streak': 9, 'max_loss_streak': 6, 'num_trades': 81}  
tr': 'Equity Curve'})
```

Equity Curve



Стартирахме етикетирането на позициите на базата на промяна от 0.08. 0.8%. Добавихме начален капитал от 10 000. През периода получихме увеличение на средствата и сортиново съотношение 1.84. Увеличихме рисковия капитал на търговия до 25% и спада беше:

Drawdown Curve (Max Drawdown: -16.13%)

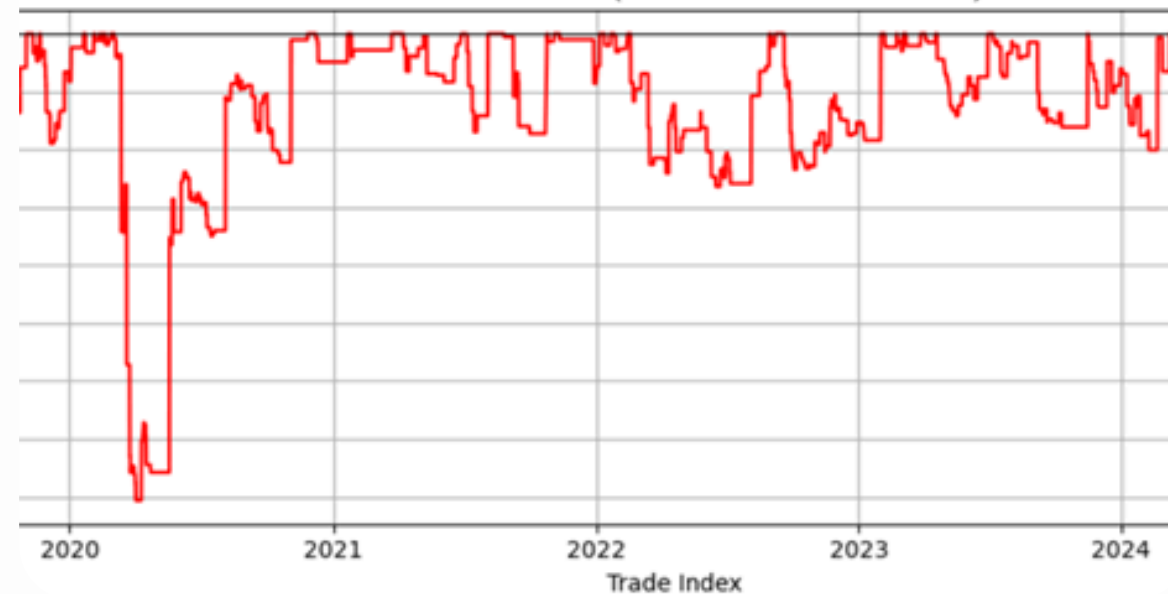


Схема срещу стратегията за купуване и задържане.

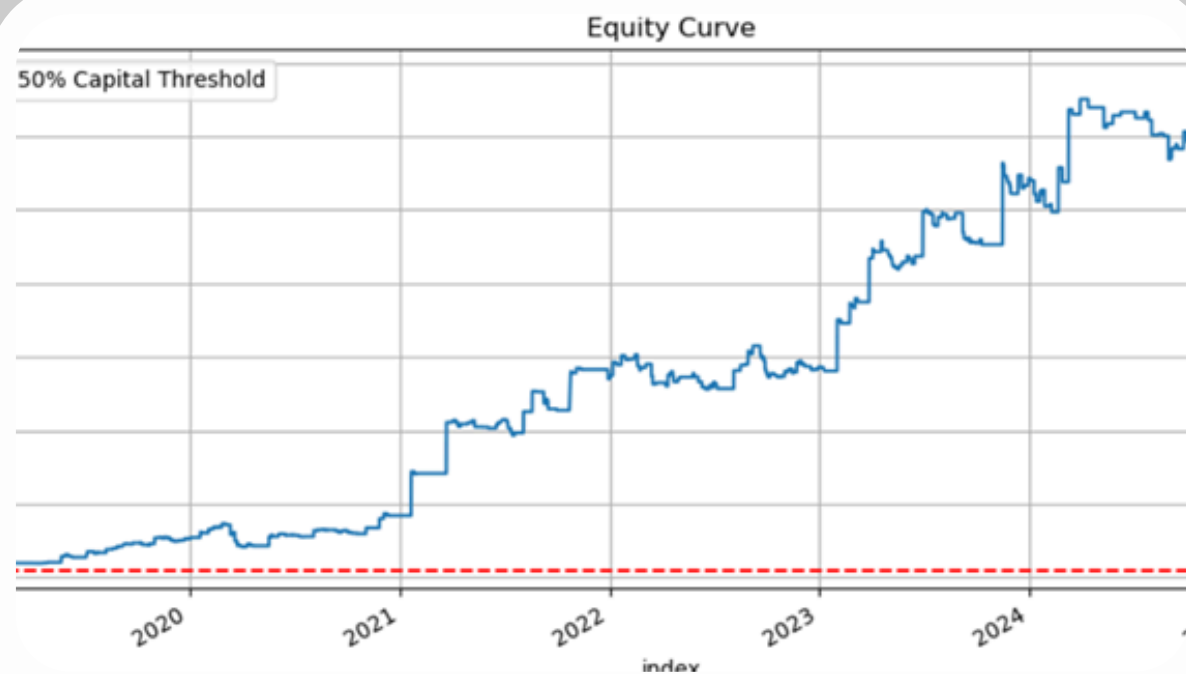
Strategy vs Buy and Hold



Ами нашият модел не се представи добре. Повечето хора твърдят, че когато инвестираш в крипто, е все едно да забравиш, че някога си имал тези пари.



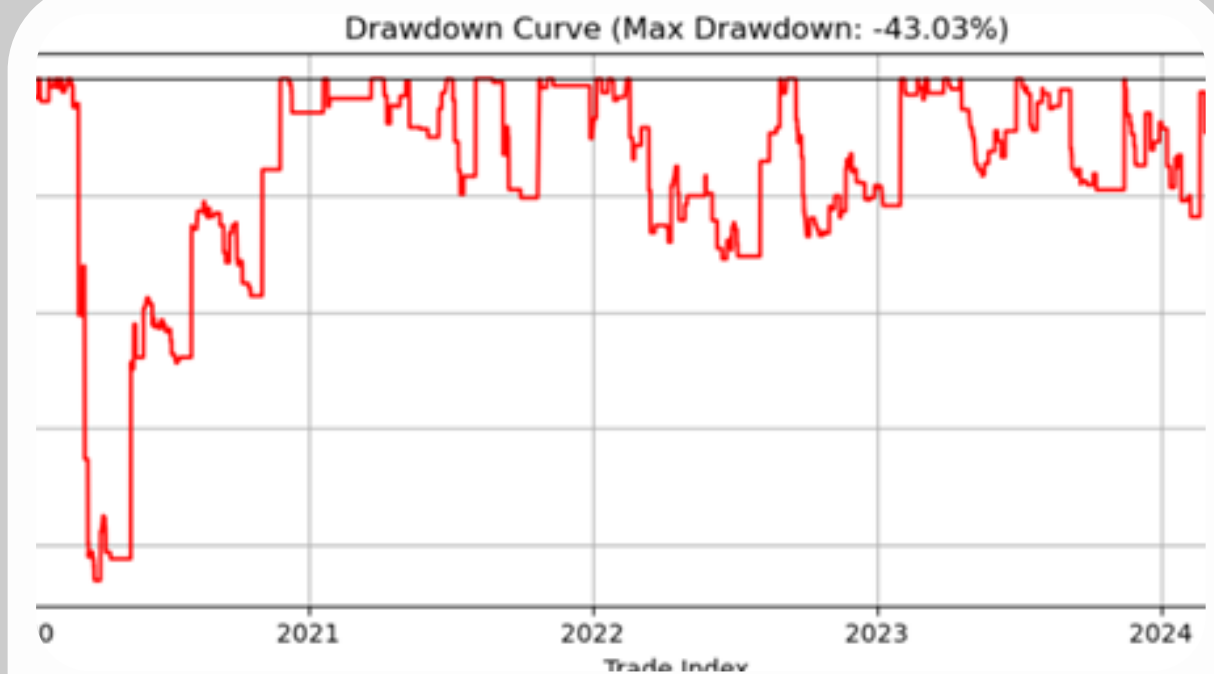
# Тестване на стратегията след повишаване на риска



Виждаме ако увеличим риска на 75 %?  
Виждаме, че сме близо, но никога не достигаме 50 %  
загуба на капитал, съответно резултатите са добри.

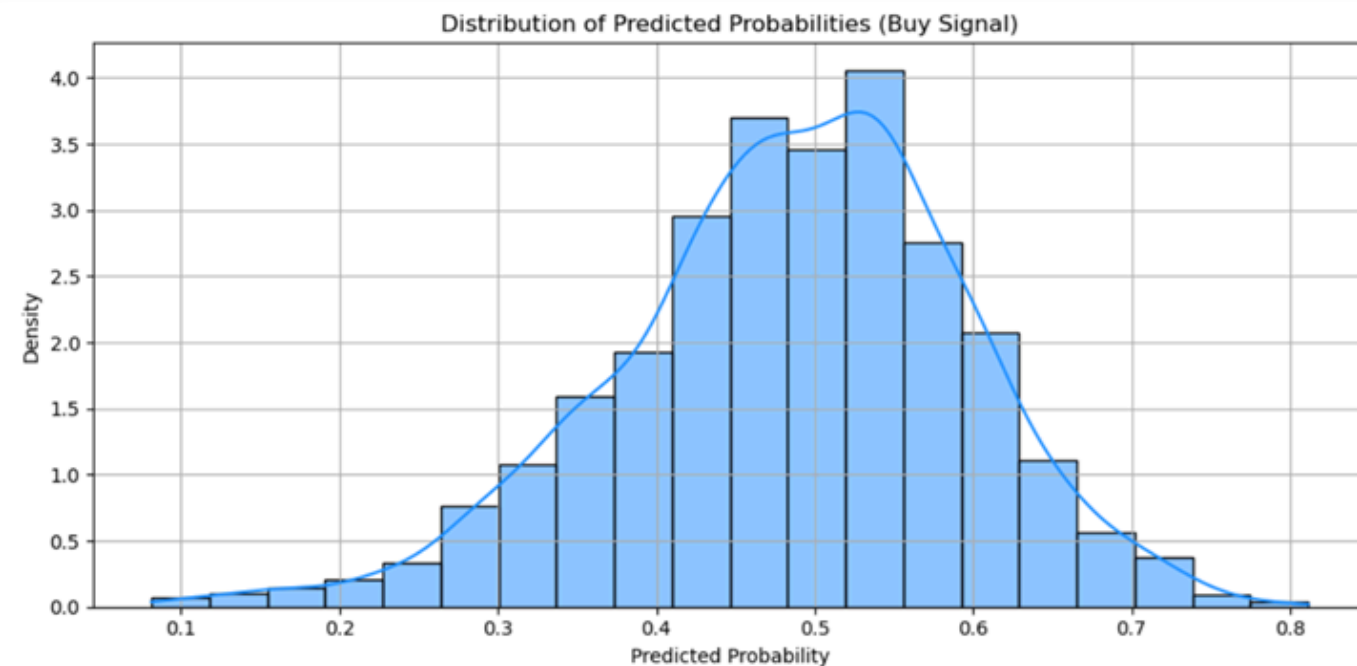


Успяхме да надминем стратегията за покупка и  
задържане малко. Виждаме, че там, където BTC  
всъщност не е добър, нашата стратегия наистина е  
печеливша.

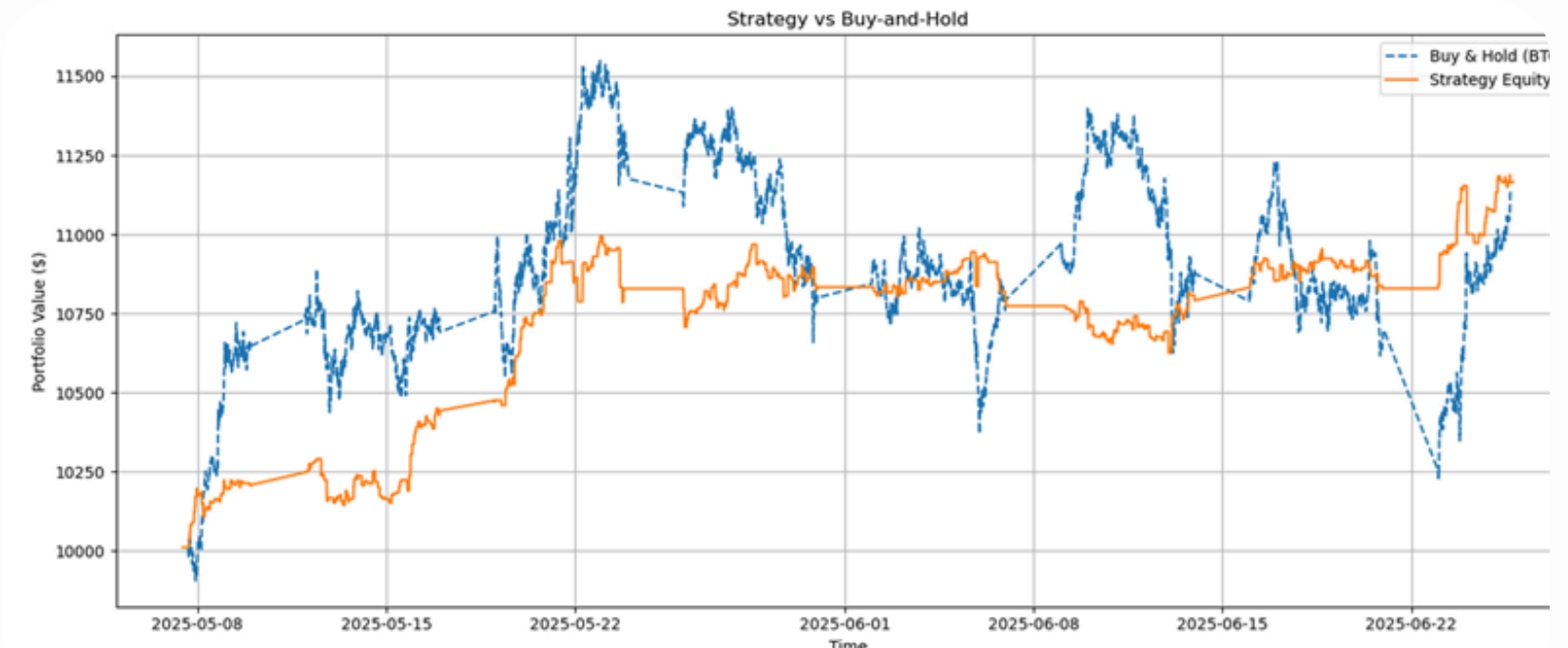


на графиката виждаме действителното намаление  
около 2020 с 43%, но не достигаме нашият лимит на  
загуба.

# Модел за бинарна класификация на база на същата архитектура на модела



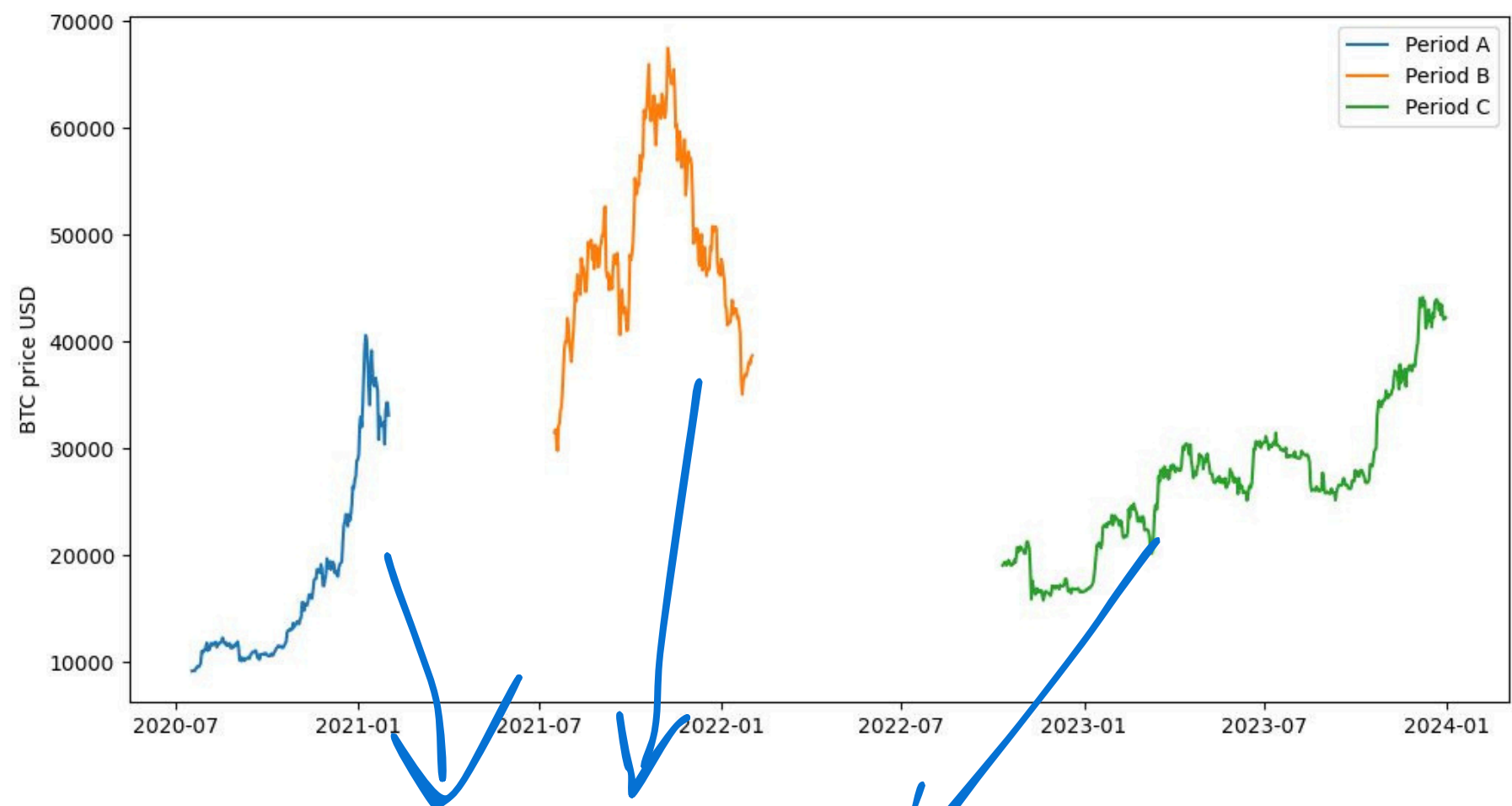
Филтрирахме вероятностите въз основа на графиката :  
така че решихме, че ако сме над 52 %, купуваме, а ако сме под 39 %, сме неутрални.



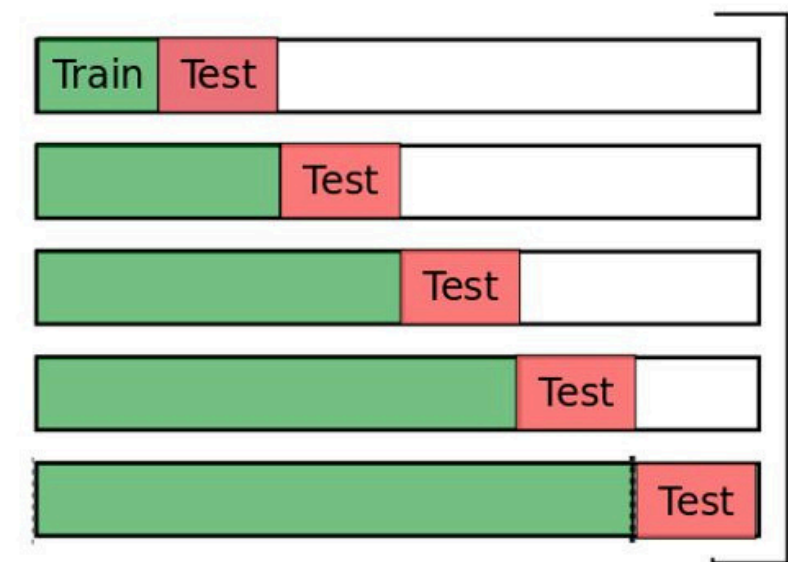
Успяхме да надминем стратегията 'купувай и задръж' с ненастроен модел, но с сделки с висок риск. Следващата ни цел е да създадем модел, който да може да надмине стратегията 'купувай и задръж' и да нямаме такъв голям риск.

# Валидация на моделите

Подаваме произволни периоди за валидация.



Валидация с един такт напред.



1. Тренираме модела.
2. Валидираме резултатите за всеки fold.
3. Осредняваме резултатите.

MAPE

MEAN ABSOLUTE PERCENTAGE ERROR

Каква е грешката в модела?

DS

DIRECTIONAL SYMMETRY

Колко често познаваме посоката на промяна?

R

STABILITY COEFFICIENT

Съотношение на средната грешка спрямо стандартното отклонение. Т.е. доколко са стабилни прогнозите ни между отделните валидационни периоди?

U

COMPUTATIONAL EFFICIENCY

Колко бързо тренираме модела и прогнозираме с него?

Z

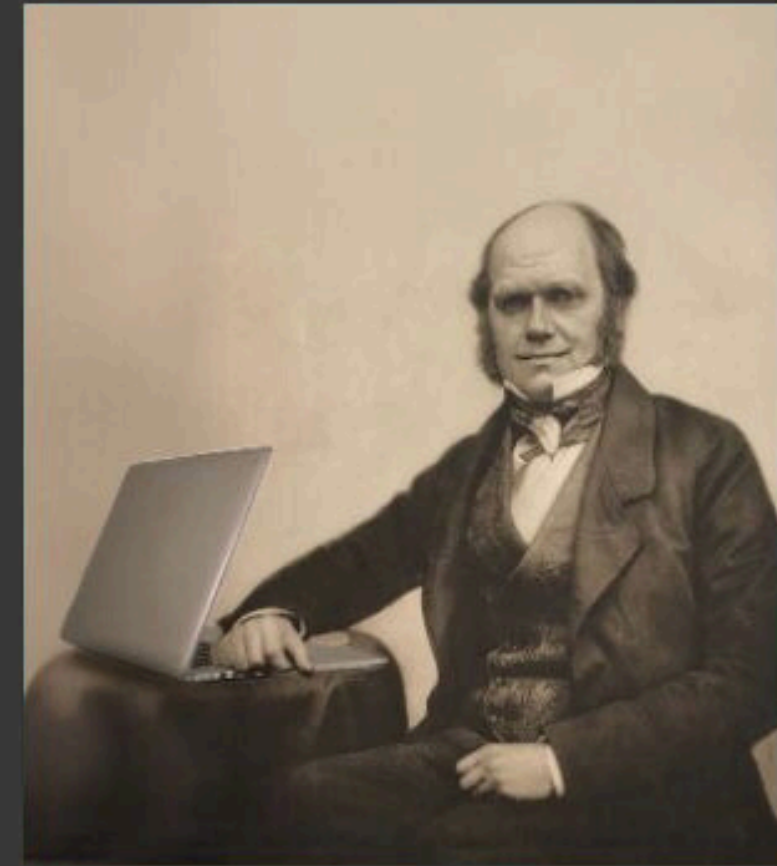
ОБОБЩЕН ИНДИКАТОР



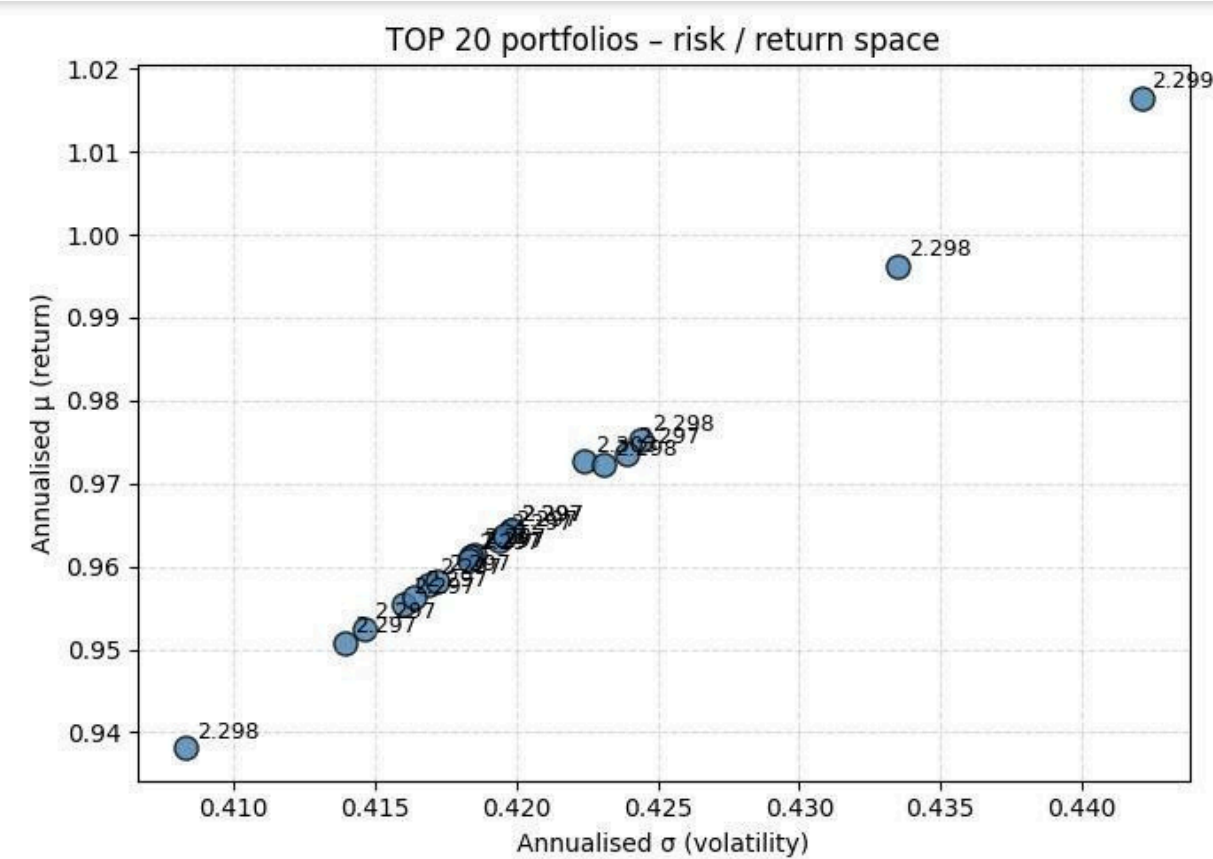
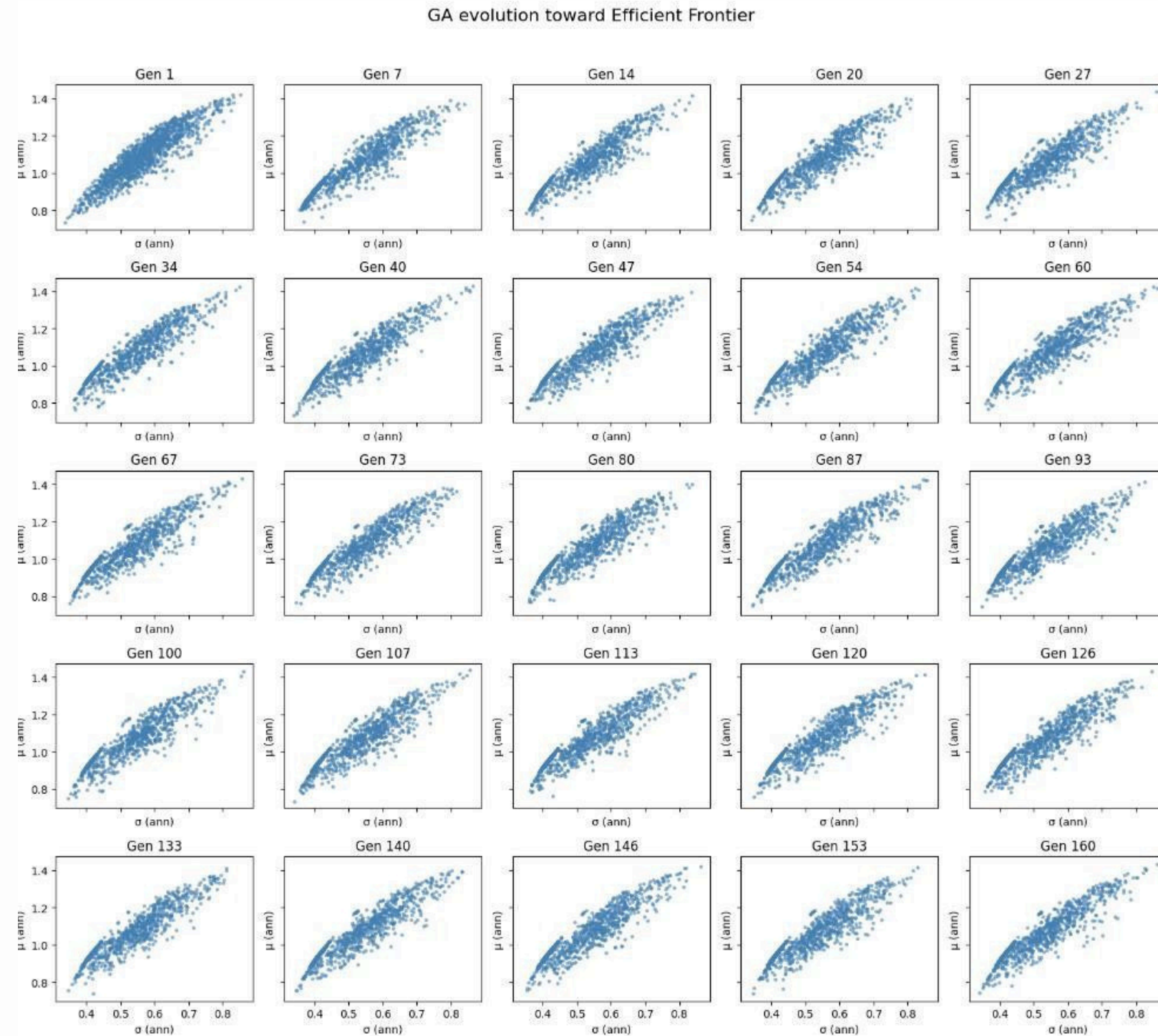
# Evolutionary Algorithms (EA)

---

- Key Idea: Inspired by natural selection.
- Main Steps:
  - a. **Initialization** of a population (random solutions).
  - b. **Evaluation (Fitness)** of each solution.
  - c. **Selection** of the best solutions.
  - d. **Recombination (Crossover)** and **Mutation** to create new solutions.
  - e. Repeat until stopping criteria are met.



# Еволюционна порфейлна оптимизация



Стартираща  
стойност на порфейла  
100 000  
Крайна  
стойност на порфейла  
219 423.78

Показването на еволюцията—как всяко поколение се приближава до границата на ефективност—е убедително доказателство, че процесът не е „на око“, а използва научен метод за търсене на стойност.



**БЛАГОДАРИМ ЗА  
ВНИМАНИЕТО!**

**Summer school**  
on Modeling, AI & Complex Systems 2025

