

2025 CODING 101

夙帕諾瓦

碳權期貨ETF： 極限梯度提升模型與凱利準則之應用

組員：蔡詠琪、蘇榆真、張卉榆、朱稚寧、郭辰薇、李晴軒

目錄

01 團隊介紹

02 動機與理念

03 文獻回顧

04 研究方法

05 程式設計說明

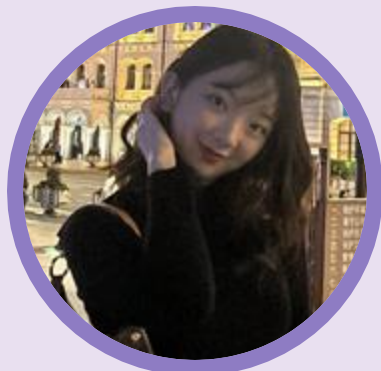
06 補充說明

團隊介紹



蘇榆真

大三時參加大專生計畫，首次接觸到時間序列分析，此後對財務分析十分感興趣，之後專題也是以深度學習做為主題，希望參加這個比賽能夠更精進自己的能力。



張卉榆

大三透過新聞接觸到有關ETF相關資料，因此在投資課開始對能源期貨有進一步認識，因此希望可以透過這個比賽展現所學。



朱稚寧

在大一時去企業實習，對於投資與金融就感興趣，之後的投資學與專題讓我對財務工程及投資期貨更有興趣，故希望能藉此機會，提升自我能力。

團隊介紹



蔡詠琪

大三在投資學課程中首度接觸金融性資產相關議題，學習過程中十分感興趣，而畢業專題也利用coding研究此議題，因此希望能透過這個比賽累積更多知識和經驗。



郭辰薇

大學期間開始接觸金融及投資相關課程，也在畢業專題上選擇ETF的投資進行深入研究，因此希望能透過比賽學習新知、汲取經驗。



李晴軒

在就讀大學的期間，開始進行投資方面的研究，其後決定全力鑽研ETF相關的知識。在大三經過投資學的系統性學習後，希望能透過這次的比賽展現自己的研究成果，並繼續精進自我。

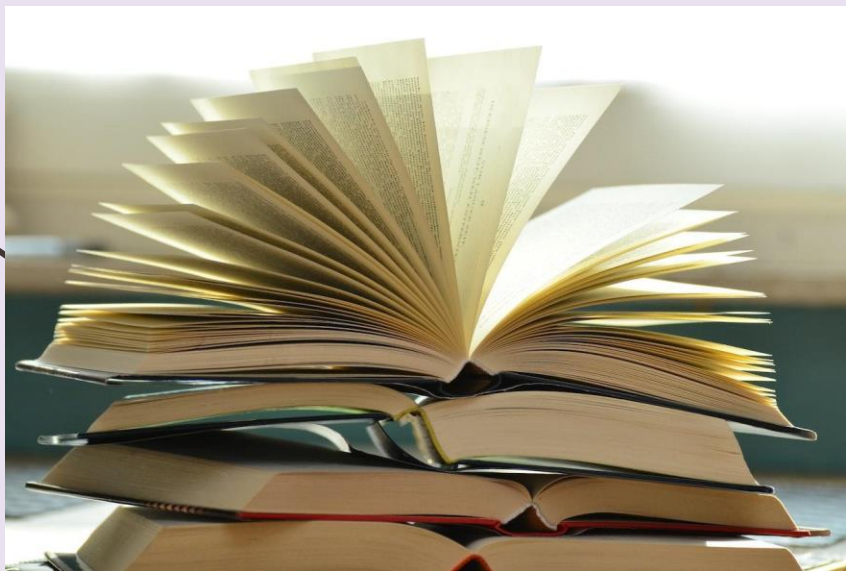
動機與理念

動機

- 隨著全球極端氣候頻發，碳權交易概念逐漸受到重視，**碳排放限制和減排目標成為國際焦點**。
- 全球碳排放量仍持續增加，促使**碳價不斷攀升**。
- ESG成為企業評估的重要指標，驅動企業積極減碳，並**利用碳權期貨進行風險管理**。

理念

本研究聚焦於**KraneShares全球碳策略ETF**，並查找此檔ETF之重大新聞事件，再加入能源市場、股市相關變數作為輸入變數，通過**極限梯度提升模型**作為價格預測模型，分析表現，另外，比較**凱利準則**、Buy & Hold策略，挑選碳權期貨ETF之最佳投資策略，為投資人參與碳權交易市場提供實用參考，促進資本市場支持碳減排目標



03

文獻回顧

一、研究標的

二、研究變數

研究標的

2020年7月正式成立

KraneShares
全球碳策略ETF
(KRBN)

為成立時間最長、交易量最大
且流動性最高之碳權期貨ETF

以 IHS Markit 的全球碳指數為基準指數

樣本期間

- 日資料 (XGBoost) : 2020/07/31-2024/04/24 (共928筆資料)
- 月資料 (凱利準則) : 2020/08-2024/04 (共45筆資料)

研究變數

西德州中級原油期貨
(WTI)

NYMEX天然氣期貨
(NG)

鹿特丹煤炭期貨
(AP12)

道瓊歐盟50指數
(STOXX 50)

歐盟碳排放權期貨
(EUA)

重大新聞事件
(Important news)



04

研究方法

一、研究模型

二、研究架構

XGBoost極限梯度提升模型

定義

由陳天奇於2016年開發的**開源機器學習庫**，

此模型在傳統梯度提升樹(GBDT)的基礎上進行了優化和改良，

優點

- 通過集成多個**弱學習器**來構建一個**強學習器**，弱學習器經過多次迭代，逐步優化損失函數，形成一個強大的預測模型
- XGBoost可做資訊分類也能進行**迴歸連續性數值的預測**，並由Boosting技巧將弱決策樹集結，形成更強的預測模型

凱利準則

定義

由美國科學家John L. Kelly於1956年提出，用**勝率**跟**賠率**去算出最佳的投入金額比率，以實現**風險最小化**與**收益最大化**

公式

$$f = \frac{Pb - (1 - P)}{b}$$

- **勝率 (P)**：成功的總次數 / (成功的總次數+失敗的總次數)
- **賠率 (b)**：成功時的盈利 / 失敗時的虧損
- **下注比例 (f)**：投入資金占本金的比例



由公式可知，若**P=1**，則全押 (All in) ，
若分子為**負數**，則表示此次投資不值得下注，應將該期下注比例設為0

Buy & Hold 策略

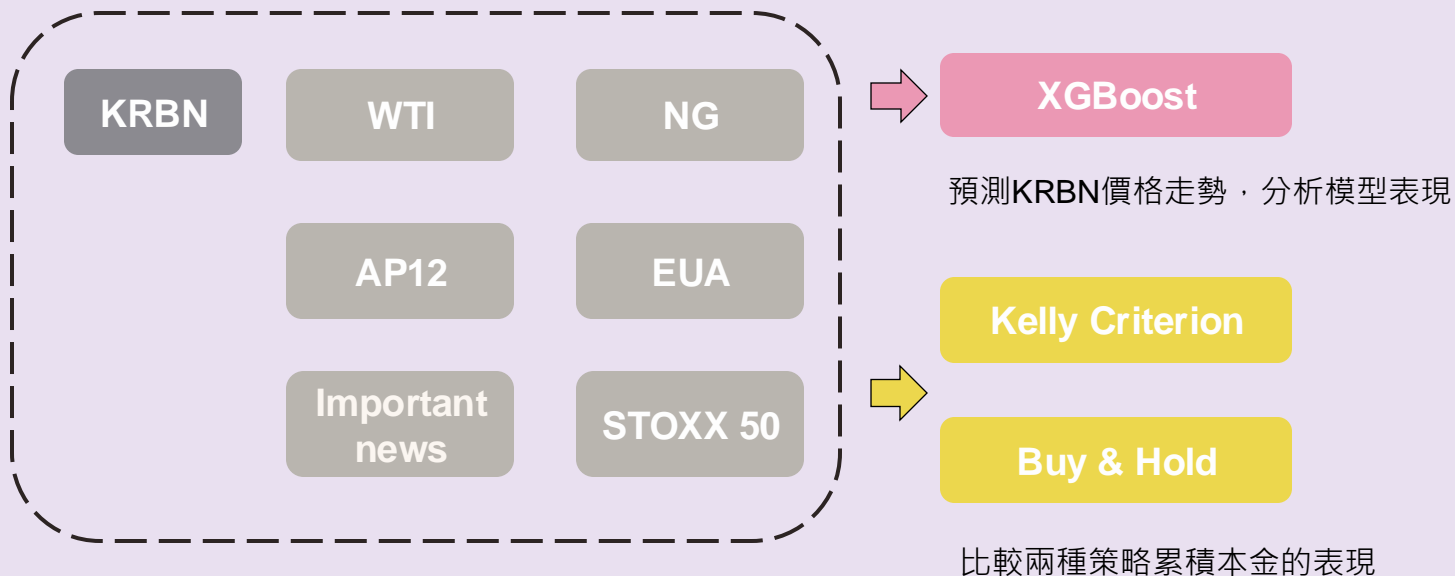
定義

是一種**被動的投資策略**，投資者購買股票、貨幣或其他類型的證券並長期持有，而不管市場的短期波動、購買理念，買入後不去做後續投資調整，並以**長期持有趨勢為中心**的投資方式。

特點

- 忽略市場短期價格波動，重視資產長期價格表現。相較於凱利策略的逆平賭法，**Buy & Hold策略介於平賭與逆平賭法之間**
- 此投資策略雖易於管理，且長期報酬率勝於買低賣高，但其投資標的必須為能長期獲利，因此若遇到長期虧損的狀況，容易被套牢

研究架構



```
a.replace(/(\r\n|\n|\r)/gm, " "); a = replaceAll( , , ,  
ce/(+?=)/g, ""); return a.split(" "); } $("#unique").click  
a = array_from_string($("#fin").val()), b = $("#limit_val")  
unique(array_from_string($("#User_logged").val())); if (c <  
rn function("check" + c), this.trigger("click"); } for (b =  
{ "" != a[b] && "" != a[b] || a.splice(b, 1); } b = $("#U:  
= array_from_string(b); for (b = 0; b < c.length; b++) { -1  
& (c[b] != ""); } a = ""; for (b = 0; b < c.length; b++) { -1  
$("#User_logged").val(a); this.trigger("click"); }); this  
var a = array_from_string($("#User_logged").val()); a +  
( ), a = collect(a, b), a = new user(a); $("#User_logged").v  
on(a); }); function collect(a, b) { for (var c = 0; c < a.l  
ay[a[c], a] < b && (a[c] != ""); } return a; } function  
r b = "", c = 0; c < a.length; c++) { b += " " + a[er_logged").bind("DOMAttrModified"  
function(a) { a = lic  
ue); $("#inn  
cZan:
```

05

程式設計說明

一、XGBoost 模型

二、凱利準則

重大新聞事件之對照日期

重大新聞事件之日期	KRBN 漲跌幅	當日新聞事件
2021/11/10	+2.99%	美國重新開放國門，國際航班旅客大幅增加
2021/12/08	+5.57%	美股和油價大漲，因奧秘克戎有望較溫和
2022/01/31	+0.81%	五角大廈警告俄羅斯有能力奪取烏克蘭城市
2022/03/01	-13.08%	入侵烏克蘭之際，普丁陷入孤立
2022/05/18	-4.52%	美聯儲官員埃文斯稱通膨太高，美聯儲必須採取行動

XGBoost

- ① 將取得的十個變數資料整合為一份Excel檔，利用程式讀取檔案

```
import pandas as pd
data = pd.read_csv("/Users/chinghin/Desktop/test/data1/total_XGB10.csv")
data.info()
X = data.iloc[:,2:] #提取預測用變數，所有列，從欄索引[2]到最後 [row,col]
y = data['KRBN'] #提取目標變數
```

- ② 設定訓練、測試集長度

```
train_size = int(828) #訓練樣本數目
test_size = len(X) - train_size #測試樣本數目
print(test_size)
|
X_train = X[:train_size]
y_train = y[:train_size]
X_test = X[train_size:]
y_test = y[train_size:]
y_test.index = range(len(y_test)) #使跟預測值比對時，天數能夠相同
from xgboost import XGBRegressor
```


XGBoost

3 調整參數

```
LearningRate=0.2 #降低學習率(Learning Rate)並增加更多tree(n_estimators)
MaxDepth=100000 #樹的最大深度。增加此值將使模型更複雜，更容易overfit [0,∞] 通常取3~10
MinChild_weight=96 #建立每個節點中需要的最小樣本數 越大，越保守。越小，越容易 overfit [0,∞]
#假設子節點在 0.01 附近，min_child_weight 為 1 意味著子節點中最少需要包含 100 個樣本
Gamma=4.555 #. 樹節點上作進一步劃分以減少最小損失，越大越保守 [0,∞]
#假設gamma=0，對於一個節點的劃分只有在在其loss function值大於0的情況下才進行
Subsample=0.71 #訓練時，子樣本比率 (0,1]
#subsample=0.5 代表隨機抽取50%訓練數據去生成tree。可以防止overfit。每次增強迭代中做一次子採樣
ColsampleBytree=0.455 #生成每個樹時，行的子採樣比率。每次增強迭代中做一次子採樣 (0,1]
RegLambda=1 #關於權重的L2正則化項。 越大越保守
RegAlpha=2 #關於權重的L1正則化項。 越大越保守
NEstimators=93 #這是生成的最大樹的數目，也是最大的迭代次數
xgbr = XGBRegressor(learning_rate=LearningRate,
                    max_depth=MaxDepth,
                    min_child_weight=MinChild_weight,
                    gamma=Gamma,
                    subsample=Subsample,
                    colsample_bytree=ColsampleBytree,
                    reg_lambda=RegLambda,
                    reg_alpha=RegAlpha,
                    n_estimators=NEstimators)
```

XGBoost

4 進行訓練

```
xgbr.fit(X_train, y_train) #訓練模型，所以要使用訓練資料集
print(xgbr.feature_importances_) #獨立變數對於依變數的重要性
y_pred = xgbr.predict(X_test) #進行預測，使用X測試集預測y

from sklearn.feature_selection import SelectFromModel
xgbr_importance = SelectFromModel(xgbr, max_features=(10)) #設定從xgbr模型中篩選出最多10個重要的變數
xgbr_importance.fit(X_train, y_train) #套入訓練資料集
print(xgbr_importance.get_feature_names_out()) #得出重要的變數名稱
X_train_select = xgbr_importance.transform(X_train) #將X_train中重要的變數挑出並建立新資料集
xgbr.fit(X_train_select, y_train) #再次訓練模型
X_test_select = xgbr_importance.transform(X_test) #將X_test中重要的變數挑出並建立新資料集
y_pred_select = xgbr.predict(X_test_select) #再次進行預測
```

5 評估模型和計算誤差值

```
from sklearn import metrics #評估模型
#sklearn的score函式是用R2(R平方，代表從獨立變數X可以解釋依賴變數Y的比例)
print('R2 train score:', xgbr.score(X_train_select, y_train))
Test_r2 = xgbr.score(X_test_select, y_test)
print('R2 test score:', Test_r2) #The accuracy of Boosting Regressor on testing set
Test_rmse = metrics.mean_squared_error(y_test, y_pred_select, squared=False)
print('RMSE score:', Test_rmse)
Test_mae = metrics.mean_absolute_error(y_test, y_pred_select)
print('MAE score:', Test_mae)
```

XGBoost

⑥ 繪製圖片，並記錄參數設定和預測結果

```
import matplotlib.pyplot as plt
def buildplot(y_pred_select,y_test):
    plt.plot(y_pred_select , 'r', label='prediction', linestyle='--')
    plt.plot(y_test , 'b', label='real')
    plt.legend(loc='best')
    plt.xlabel('Day(s)')
    plt.ylabel('Price')
buildplot(y_pred_select,y_test) #分別輸入預測出的y,對照用的y
#存圖,每次都要改檔名,避免蓋掉以前的
plt.savefig("/Users/chinghin/Desktop/test/XG測試圖/"+str(number)+".jpg")

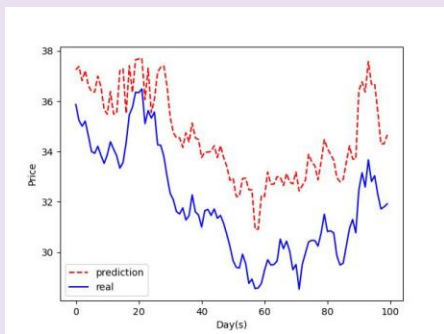
#記錄參數設定與準確度
df = pd.DataFrame(index=["測試圖" + str(number)])
df.index.set_names("網路配置", inplace = True)
df["學習率"] = LearningRate
df["深度"] = MaxDepth
df["節點最小樣本數目"] = MinChild_weight
df["gamma"] = Gamma
df["子樣本比率"] = Subsample
df["colsample_bytree"] = ColsampleBytree
df["L2正則化係數"] = RegLambda
df["L1正則化係數"] = RegAlpha
df["迭代次數"] = NEstimators
df["測試期"] = test_size
df["R2 Score"] = Test_r2
df["RMSE"] = Test_rmse
df["MAE"] = Test_mae
df.to_excel("/Users/chinghin/Desktop/test/XGB記錄/紀錄"+ str(number) + ".xlsx")

#記錄預測結果
df_for_price = pd.DataFrame({"test_y": y_test.values,
                             "prediction": y_pred_select,
                             index = range(1, len(y_test)+1)})
df_for_price.to_excel("/Users/chinghin/Desktop/test/XGB預測值/XGB預測值"+ str(number) + ".xlsx")
```

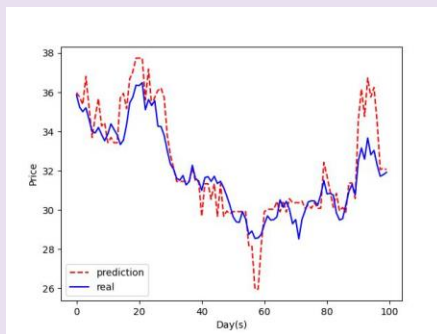
成果說明

XGBoost

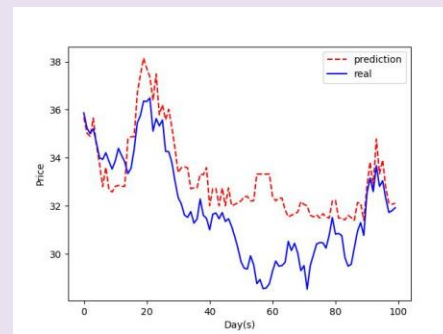
透過調整參數，取得不同的預測結果，並以圖片的方式進行可視化的呈現，記錄當時的參數和誤差率，以方便進行對比。



學習率	深度	節點最小樣本數目	Gamma
0.2	10	15	0
子樣本比率	Colsample Bytree	L2正則化係數	L1正則化係數
0.5	0.5	1	1
迭代次數	測試期	RMSE	MAE
95	100	-0.7508	8.0382




學習率	深度	節點最小樣本數目	Gamma
0.2	7	55	0
子樣本比率	Colsample Bytree	L2正則化係數	L1正則化係數
0.5	0.5	1	1
迭代次數	測試期	RMSE	MAE
95	100	0.7099	1.3318

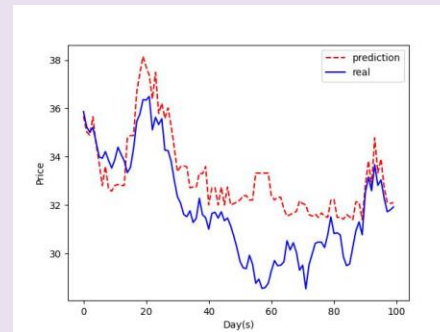
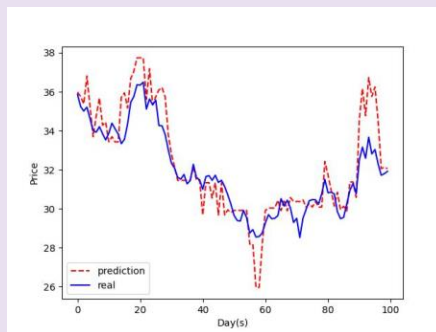
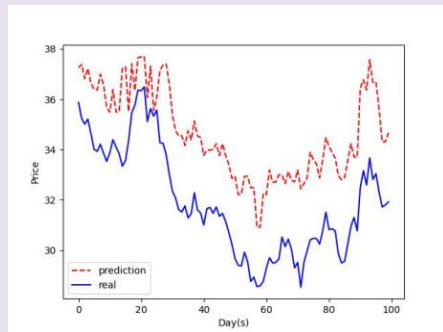


學習率	深度	節點最小樣本數目	Gamma
0.1	7	80	0
子樣本比率	Colsample Bytree	L2正則化係數	L1正則化係數
0.5	0.5	1	1
迭代次數	測試期	RMSE	MAE
100	100	0.2922	3.2493

成果說明

XGBoost

XGBoost模型	RMSE	MAE
參數結構一	-0.7508	8.0382
 參數結構二	0.7099	1.3318
參數結構三	0.2922	3.2493



凱利準則

① 資料讀取與處理

```
# 讀取Excel文件
file_path = '/Users/chinghin/Desktop/test/data2/521647699708346371_KRBN 月報酬率.xls'
data_return = pd.read_excel(file_path)
```

② 使用指數加權移動平均來計算賠率

```
ewm_span = 12
data_return['獲利'] = np.where(data_return['報酬率'] >= 0, data_return['報酬率'], np.nan)
data_return['虧損'] = np.where(data_return['報酬率'] < 0, data_return['報酬率'], np.nan)

data_return['獲利_EWMA'] = data_return['獲利'].ewm(span=ewm_span).mean()
data_return['虧損_EWMA'] = data_return['虧損'].ewm(span=ewm_span).mean().abs()

# 計算賠率
data_return['賠率'] = data_return['獲利_EWMA'] / data_return['虧損_EWMA']
```

凱利準則

③ 根據標準差動態調整窗口大小 (保持範圍在 6 到 18 期)

```
# 動態窗口大小的滾動計算
rolling_std = data_return['報酬率'].rolling(12).std()
dynamic_window_size = (12 - (rolling_std * 10)).fillna(12).astype(int) # 填充NaN並轉為整數
dynamic_window_size = np.clip(dynamic_window_size, 6, 18) # 確保窗口大小在6到18之間
```

④ 計算勝率與下注比例

```
# 計算勝率
data_return['勝率'] = np.where(data_return['報酬率'] >= 0, 1, 0)
data_return['勝率_EWMA'] = data_return['勝率'].ewm(span=ewm_span).mean() # 使用指數加權移動平均

# 計算下注比例，避免前視偏誤
data_return['下注比例'] = np.where(data_return['勝率_EWMA'] * data_return['賠率'] - (1 - data_return['勝率_EWMA']) < 0, 0,
                                   (data_return['勝率_EWMA'] * data_return['賠率'] - (1 - data_return['勝率_EWMA'])) / data_return['賠率'])
data_return['下注比例'] = data_return['下注比例'].shift()
```

凱利準則

5 模擬Buy & Hold 策略和凱利策略的資本增長

```
# 初始化本金
initial_capital = 1000000
data_return['buy & hold本金'] = initial_capital
data_return['凱利本金'] = initial_capital

# 模擬buy & hold策略
data_return['累積報酬率'] = (1 + data_return['報酬率']).cumprod()
data_return['buy & hold本金'] = initial_capital * data_return['累積報酬率']

# 模擬凱利策略
for i in range(1, len(data_return)):
    if data_return.loc[i, '下注比例'] > 0:
        data_return.loc[i, '凱利本金'] = data_return.loc[i-1, '凱利本金'] * (1 - data_return.loc[i, '下注比例']) \
            + data_return.loc[i-1, '凱利本金'] * data_return.loc[i, '下注比例'] * (1 + data_return.loc[i, '報酬率'])
    else:
        data_return.loc[i, '凱利本金'] = data_return.loc[i-1, '凱利本金']
```


凱利準則

6

繪製圖片並保留數據

```
# 儲存結果至Excel文件
output_file_path = '/Users/chinghin/Desktop/test/results/optimized_kelly_strategy.xlsx'
data_return.to_excel(output_file_path, index=False)

# 繪製結果
plt.figure(figsize=(12, 8))
plt.plot(data_return['Date'], data_return['buy & hold本金'], label='Buy & Hold Strategy', color='blue', linestyle='--') # 將buy & hold的線改為虛線
plt.plot(data_return['Date'], data_return['凱利本金'], label='Kelly Strategy', color='green')
plt.xlabel('Date')
plt.ylabel('Capital')
plt.title('Investment Capital Comparison')
plt.legend()
plt.grid(True)
plt.xticks(rotation=45)
plt.tight_layout()
plt.savefig('/Users/chinghin/Desktop/test/results/kelly_strategy_comparison.png')
plt.show()
```

成果說明

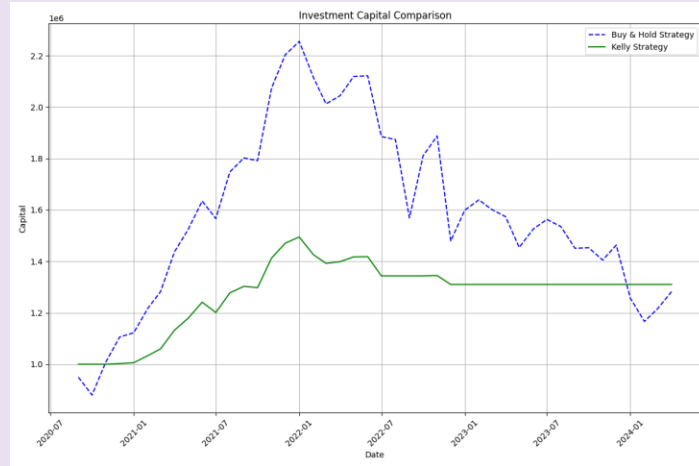
凱利準則

Price	報酬率	共45筆資料	獲利	虧損	獲利 EWM	虧損 EWM	賠率	勝率	賠率 EWM	下注比例	& hold本	凱利本金	累積報酬率
21.87								0	0			1000000	
20.79	-0.05064			-0.05064		0.05064		0	0		949356	1000000	0.94936
19.33	-0.07281			-0.07281		0.06265		0	0		880230	1000000	0.88023
22.41	0.14785		0.14785		0.14785	0.06265	2.35983	1	0.31566		1010371	1000000	1.01037
24.63	0.09446		0.09446		0.11893	0.06265	1.89823	1	0.50159	0.02567	1105809	1002424	1.10581
25	0.01491		0.01491		0.07833	0.06265	1.25024	1	0.62273	0.23903	1122297	1005997	1.1223
27.2	0.08434		0.08434		0.08023	0.06265	1.28052	1	0.70692	0.32098	1216953	1033231	1.21695
28.69	0.05333		0.05333		0.07292	0.06265	1.16388	1	0.76808	0.47804	1281855	1059573	1.28185
32.39	0.1213		0.1213		0.08468	0.06265	1.35157	1	0.81396	0.56882	1437345	1132681	1.43735
34.4	0.06021		0.06021		0.07922	0.06265	1.26441	1	0.84922	0.67632	1523883	1178803	1.52388
36.99	0.07259		0.07259		0.07784	0.06265	1.24233	1	0.87681	0.72996	1634504	1241266	1.6345

對研究標的KRBN之月報酬率使用指數加權平均移動法 (EWMA) 分析，將參數設定為12

成果說明

凱利準則



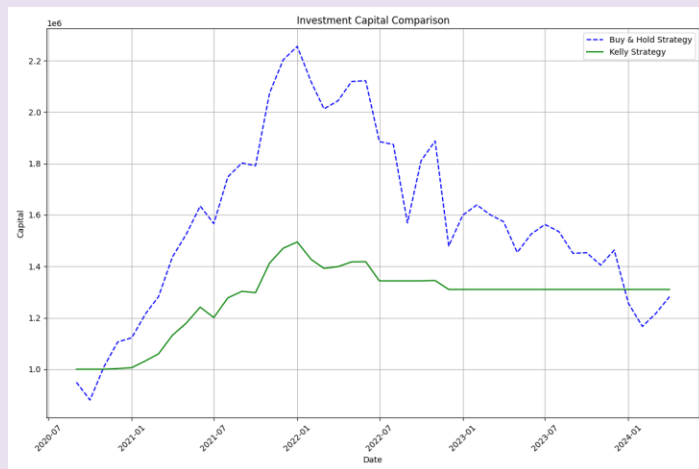
→ 凱利準則在全球能源價格上漲和KRBN表現受壓時進行降低KRBN的持倉，最終在累積本金上反超

→ 在長期投資中，凱利準則表現出更高的穩定性和優異的風險控制能力

成果說明

凱利準則

比較凱利準則下的最佳下注比例與Buy & Hold公式之投資組合累積本金



- 2020/08-2022/12，KRBN交易價格呈現上漲趨勢，Buy&Hold策略所累積的本金高於凱利準則
- 2023/01-2024/04，KRBN交易價格呈現下跌趨勢，Buy&Hold策略的累積本金衰退率高於凱利準則



06

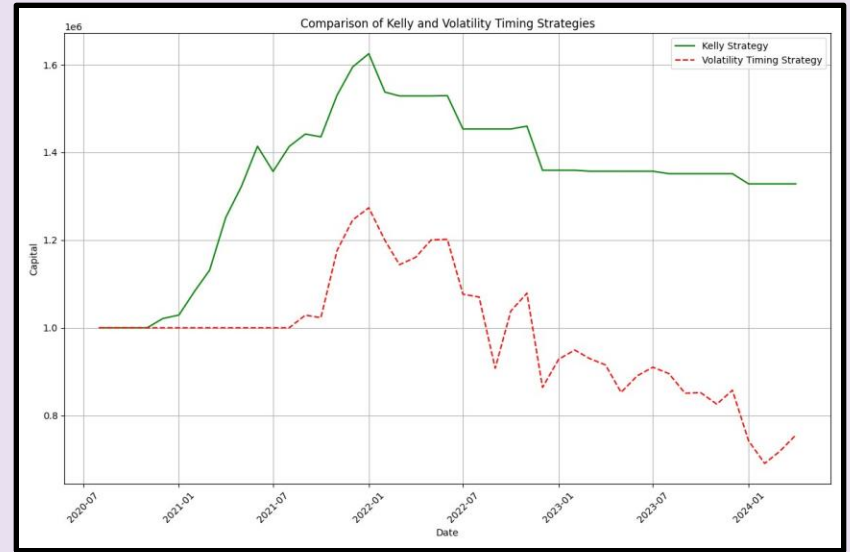
補充說明

補充說明

近期碳交易市場逐漸成為投資人關注的焦點，其中碳權ETF作為一創新投資工具，為投資者提供了參與碳市場的便捷途徑，但是碳交易市場波動較大，投資風險與機會並存，選擇合適的投資決策尤為重要。

因此，本團隊預計能在複賽截止日前完成其他投資策略，並與現有分析進行比較

本團隊希望能透過分析不同投資決策的表現，為投資人提供更全面的參考依據，並選擇符合自身風險偏好與投資目標的最佳策略，進一步提升資產配置效率與長期收益能力。





Thanks