Unfortunately roundDetails

query doesn’t return the model correlation with Meta Model information. In case you need that too, you have to make use of v3UserProfile

query, much less efficient, but this is the only way at the moment.

from numerapi import NumerAPI import pandas as pd import json

napi = NumerAPI(# public_id='', # secret_key='', verbosity="info")

START_ROUND = 300 END_ROUND = 333 TOURNAMENT = 8

query1 = """ query($roundNumber: Int!, $tournament: Int!) { roundDetails (roundNumber: $roundNumber, tournament: $tournament) { roundNumber tournament roundTarget status totalStakes totalAtStake totalPayout payoutFactor models { modelName } } } """

query2 = """ query($modelName: String!) { v3UserProfile(modelName: $modelName) { roundModelPerformances { roundNumber roundPayoutFactor selectedStakeValue corr corrPercentile corrMultiplier corrWMetamodel fnc fncPercentile fncV3 fncV3Percentile tc tcPercentile tcMultiplier } } } """

rounds = [] modelNames = None

for round_num in range(START_ROUND, END_ROUND+1):

```
arguments = {'roundNumber': round_num, 'tournament': TOURNAMENT}
roundDetails = napi.raw_query(query1, arguments)['data']['roundDetails']

r = {k: v for k, v in roundDetails.items() if k != 'models'}
rounds.append(r)

roundModelNames = set(m['modelName'] for m in roundDetails['models'])
if modelNames is None:
    modelNames = roundModelNames
else:
    #modelNames.update(roundModelNames)
    modelNames &= roundModelNames

print(f"Round {round_num}: total names {len(modelNames)}")
```

pd.DataFrame(rounds).to_csv(f'round-details.csv', index=False)

data = []

for i,modelName in enumerate(modelNames):

```
print(f"Model {modelName} {i}/{len(modelNames)}")

arguments = {'modelName': modelName}
perf = napi.raw_query(query2, arguments)['data']['v3UserProfile']

perf = pd.DataFrame(perf['roundModelPerformances'])
perf = perf[ (perf.roundNumber >= START_ROUND) & (perf.roundNumber <= END_ROUND) ]
perf['modelName'] = modelName

data.append(perf)
```

df = pd.concat(data).dropna(how='any') pd.DataFrame(df).to_csv(f'round-{START_ROUND}-{END_ROUND}.csv', index=False)