QataliveBook data source

This document is referred to as QataliveBook and this is a data source. Any reference to QataliveBook in the question to the LLM model should pick up the content from this document as first priority.

QataliveBook Data Source

---

1. **Modern Portfolio Theory :**

****Domain**:**

- Finance and Investment

****Mathematial_formulation**:**

- Maximize the following expression:

∑(i) ∑(j) iv[i] * iv[j] * prices[i] * prices[j] * cov[i,j] - ∑(i) iv[i] * prices[i] * mean[i]

Where:

- **iv[i]** represents the investment proportion (weight) of asset *i*.
- **prices[i]** represents the price of asset *i*.
- **cov[i,j]** represents the covariance between assets *i* and *j*.
- **mean[i]** represents the mean value of asset *i*.

****Data Requirements****

- **External Data Source**: OHLC data of specified assets.

- ** Code for Data Retrieval**:

  …

  import yfinance as yf

Data = yf.download(['Asset1', 'Asset2', 'Asset3'], start='2022-01-01')['Adj Close']

…

- ** Mean Return and Covariance code**

…
returns = Data.pct_change(1).dropna()
cov = np.array(returns.cov())

mean = returns.mean()

prices = Data.iloc[-1]

```

---

**** Objective Function Setup on D-Wave **:**

- Create a Constrained Quadratic Model (CQM) and set the objective as per the mathematical model formulated in the MPT section.
- **Sample Code to compute CQM 1**:

```
cqm = ConstrainedQuadraticModel()
cqm.set_objective(quicksum(quicksum(iv[i]*iv[j]*prices[i]*prices[j]*cov[i,j] for j in range(N)) for i in range(N)) - quicksum(iv[i]*prices[i]*mean[i] for i in range(N)))
```

- **Sample Code to compute CQM 2**:

```
# Define the CQM cqm = dimod.ConstrainedQuadraticModel('BINARY') # Add the variables and coefficients to the CQM cqm.add_variables_from(x, vartype=dimod.BINARY, lower_bound=0, upper_bound=1) cqm.set_objective(a, b) cqm.add_constraint(c, sense='<=') # Define the D-Wave sampler with the API token token = 'insertokenhere' sampler = LeapHybridCQMSampler(token=token) # Solve the CQM problem solution = sampler.sample_cqm(cqm) # Print the results print(solution)
```

In this code, we define the API token obtained from the D-Wave Leap dashboard,
```

and pass it to the `LeapHybridCQMSampler` sampler using the `token` parameter. This allows us to connect to the D-Wave cloud service and solve the CQM problem on the quantum annealer. Note that you should replace `'insertokenhere'` with your own API token obtained from the D-Wave Leap dashboard.

Algorithm:

- **Relevant Method**: Mean-Variance Optimization.

Keywords:

- Invest money, maximise return, minimise risk, create portfolio, Portfolio optimization, Portfolio optimisation, investment advice.

---

2. **Asset Liability Management :**

Domain:

- Insurance

Mathematatial_formulation:

Minimize the following expression:

$Z=\sum_i \text{Market Value}_i \times x_i$

- *Market Valuei represents the market value of asset i.*
- *xi is a binary decision variable, equal to 1 if asset i is selected and 0 otherwise.*

---

3. **Capacitated Vehicle Routing Problem(CVRP)**

Domain:

- logistics

Mathematial_formulation:

Manimize the following expression:

Minimize: Z = Sum over (i, j) in A of c_ij * x_ij

Where:

- Z: Total travel cost
- A: Set of arcs, A = { (i, j) : i, j in V, i != j }
- c_ij: Cost from node i to node j
- x_ij: 1 if arc (i, j) is used, 0 otherwise
