Delta Calculation Report

Inputs and outputs

Inputs:

- the ticker of the underlying asset
- target future price in USD
- expected date
- maximum risk in USD
- interest rate
- ranking basis
- delta range

input example:

```
input_dict = {
    "Ticker": "WMT",  # Ticker of underlying asset;
    "Target Price": 135.0,  # Estimated future price
    "Target Date": "2020-08-21",  # Estimated time when the price hit
target price
    "Maximum Risk": 5000.0,
    "Contract Number": 5,  # Number of different options contracts to
be recommended
    "Interest Rate": 1.0,  # interest rate used for delta calculation
    "Rank": "Delta",  # 'Return' or 'Delta'
    "Delta Range": (0.4, 0.5)
}
```

Outputs:

- contract type (call/put)
- strike price
- expiration date
- No. of contracts
- entry cost
- delta
- estimated return at target price

output example:

Long Call: Buy 15.0 x 21 Aug $132.0 Call @ $3.2
Delta: 0.5
Entry cost: $4800.0
Maximum risk: $4800.0
Est. return at target price: $-300.0 (-6.25%)
==========
Long Call: Buy 12.0 x 04 Sep $133.0 Call @ $4.1
Delta: 0.48
Entry cost: $4920.0
Maximum risk: $4920.0
Est. return at target price: $-2520.0 (-51.22%)
==========
Long Call: Buy 15.0 x 28 Aug $133.0 Call @ $3.3

Delta: 0.47
Entry cost: $4950.0
Maximum risk: $4950.0
Est. return at target price: $-1949.8 (-39.39%)
==========
Long Call: Buy 18.0 x 21 Aug $133.0 Call @ $2.7
Delta: 0.45
Entry cost: $4860.0
Maximum risk: $4860.0
Est. return at target price: $-1260.2 (-25.93%)
==========
Long Call: Buy 13.0 x 04 Sep $134.0 Call @ $3.7
Delta: 0.44
Entry cost: $4810.0
Maximum risk: $4810.0
Est. return at target price: $-3509.9 (-72.97%)
==========

Steps

Step1

The most recent stock price is retrieved from python yfinance library

```
current_price = yf.Ticker(dic["Ticker"]).history("1d")["Close"].iloc[-1]
```

Step2

Compare the target price and current stock price and decide the type of the contract(call/put).

Step3

The options information (strike price, expiration date, implied volatility, and price) will be scraped from Yahoo Finance.

Example link: https://finance.yahoo.com/quote/WKHS/options?p=WKHS&date=1597363200

Step4

If the ranking is based on delta, the deltas of all the contracts are calculated using "mibian" python library. Inputs needed are strike price, current price, interest rate, remaining days, and implied volatility.

```
def calculate_delta(dic, df, contract_type, curr_price):
    df["Remaining Days"] = df["Expiration Date"] - datetime.now()
    if contract_type == "Call":
        delta_ls = [
            mibian.BS(
                [curr_price, row["Strike"], dic["Interest Rate"],
row["Remaining Days"].days],
                row["Volatility"][:-1]
            ) .callDelta
            for index, row in df.iterrows()
        ]
    elif contract_type == "Put":
        delta_ls = [
            mibian.BS(
                [curr_price, row["Strike"], dic["Interest Rate"],
row["Remaining Days"].days],
                row["Volatility"][:-1]
            ).putDelta
            for index, row in df.iterrows()
        ]
    else:
        raise ValueError
    df["Delta"] = pd.Series(delta_ls, index=df.index)
    return df
```

Step5

No. of contracts will be calculated based on the maximum risk amount and price of the option. Estimated return will be also calculated.

Step6

The option contracts are sorted according to either return or delta. The option contracts are also filtered according to the delta range provided.

```
if dic["Rank"] == "Return":
    # Sort the dataframe according to the estimated return.
    df = df.sort_values(by="Estimated Return", ascending=False)
elif dic["Rank"] == "Delta":
    df = df[df["Delta"] >= dic["Delta Range"][0]]
    df = df[df["Delta"] <= dic["Delta Range"][1]]
    if len(df) == 0:
        return pd.DataFrame()
    if contract_type == "Call":
        df = df.sort_values(by="Delta", ascending=False)
    if contract_type == "Put":
        df = df.sort_values(by="Delta")
return df[:dic["Contract Number"]]
```