```
import pandas as pd
import numpy as np

# Load the data from the uploaded CSV file
data = pd.read_csv('Data.csv', sep=';')

# Define parameters
n_forecast = 30  # number of forecast periods
n_vintage = data.shape[0]  # number of vintages
discount_rate = 0.025  # annual discount rate (2.5%)

# Extract historical data
historical_cf = data.iloc[:, 2:].values  # historical cash flows
amount_originated = data.iloc[:, 1].values  # amounts originated per vintage

# Calculate the number of periods remaining for forecast per vintage
periods_remaining = n_forecast - np.arange(1, n_vintage + 1)

# Compute the historical repayment percentages
paid_percentages = historical_cf / amount_originated[:, None]  # repayment percentage per vintage

# Initialize the repayment percentages for the forecast periods
first_period = np.diag(paid_percentages)  # repayment percentage in the period loans were originated
second_period = np.concatenate([
    np.diag(paid_percentages[:-1, 1:]),
    np.array([paid_percentages[-1, -1] * 2])
])

# Initialize the matrix for forecast repayment percentages
p = np.zeros((n_vintage, n_forecast))
```

```
p[:, 0] = first_period  # assign repayment percentage of first period
p[:, 1] = second_period  # assign repayment percentage of second period

# Calculate expected repayment percentages for future periods
for i in range(n_vintage):
    for j in range(2, n_forecast):
        p[i, j] = max(0, p[i, 1] * np.log(1 + (1 - np.sum(p[i, :j]))) * (1 - (j) / n_forecast))

# Matrix for forecasted repayment percentages (for periods starting January 2021)
p_forecast = np.zeros((n_vintage, n_forecast - 1))
for i in range(n_vintage):
    for j in range(periods_remaining[i]):
        p_forecast[i, j] = p[i, n_forecast - periods_remaining[i] + j]

# Apply discount factors (monthly discount rate)
discount_factors = 1 / (1 + discount_rate) ** (np.arange(1, n_forecast) / 12)

# Calculate the present value of forecasted cash flows
pv = (p_forecast * discount_factors) * amount_originated[:, None]

# Compute the portfolio value (sum of present values of forecasted cash flows)
result = np.sum(pv)

# Print the portfolio value
print(f"The fair value estimate for the portfolio is {round(result, 2)} Swiss Francs")

# Compare with the client's estimate
client_estimate = 84993122.67  # client's estimated portfolio value
absolute_difference = abs(result - client_estimate)
relative_difference = (absolute_difference / client_estimate) * 100
```

```
# Print the difference
print(f"Absolute difference: {round(absolute_difference, 2)} Swiss Francs")
print(f"Relative difference: {round(relative_difference, 2)}%")

# Check if the difference is within the acceptable threshold
if absolute_difference < 500000:
    print("The difference is within the acceptable threshold of CHF 500,000.")
else:
    print("The difference exceeds the acceptable threshold.")
```