

Lilium Business Model

lgd, MGpai
liliumergo@protonmail.com

March 8, 2023

1 Lilium Setup Fee

We want to derive a function for the setup-fee of our service. Given the constraints of our business, we determined that the fee could be modeled using the natural logarithm function, parameterized by two coefficients α and β . In general, we seek to obtain an equation of the following form, where we take the floor for integer fees:

$$\phi(x) = \lfloor \alpha \ln(\beta x + k) \rfloor, \quad x \geq 0 \quad (1)$$

Since we want the function to simulate reality, we impose that $k = 1$ such that $\phi(0) = 0$. We are therefore left with the following equation:

$$\phi(x) = \lfloor \alpha \ln(\beta x + 1) \rfloor, \quad x \geq 0 \quad (2)$$

In order to determine the coefficients α, β we use a SciPy curve-fit model, with the following initial conditions:

$$(x_0, \phi_0) = (0, 0) \quad (3)$$

$$(x_1, \phi_1) = (100, 75) \quad (4)$$

$$(x_2, \phi_2) = (500, 150) \quad (5)$$

The Python code is provided in the appendix, and the model yielded the following values:

$$\alpha \approx 54.10 \quad (6)$$

$$\beta \approx 0.03 \quad (7)$$

2 Lilium Service Fee

During each NFT sale transaction, a flat-rate fee of 5% will be charged. This will apply to everyone, equally. However, the sale creator will have the option of having this fee taken from, or added onto, their original NFT price.

3 Appendix

```
[12]: import numpy as np
import matplotlib.pyplot as plt
from scipy.optimize import curve_fit

[13]: # The model for the setup-fee function
# It follows a natural logarithm with three parameters,
# where c = 1 such that (0,0) is a valid point.
def model(x, a, b):
    return a*np.log(b*x + 1)

[14]: # Determining the optimal parameters
# There are three points we want to have in the model:
# (x0, y0) = (0, 0)
# (x1, y1) = (100, 75)
# (x2, y2) = (500, 150)
xdata = np.array([0, 100, 500])
ydata = np.array([0, 75, 150])
params, errors = curve_fit(model, xdata, ydata, bounds=(0, [500, 1]))
print("Params: ", params[0], params[1])
print("Error: ", errors[0], errors[1])

Params:  54.10106403333622  0.0300000000000000025
Error:  [ 1.62425646e-24 -2.27388504e-27] [-2.27388504e-27  3.32950905e-30]

[15]: # We plot the floor of the model function and make it look pretty
plt.title("Lilium Setup-Fee Function")
plt.xlabel("NFT Amount")
plt.ylabel("Setup-Fee (USD)")
plt.ylim(0, 200)
plt.xlim(0, 1000)
x = np.linspace(0, 15000, 10000)
plt.plot(x, np.floor(model(x, *params)), color="Purple")
plt.savefig("lilium-setup-fee-function-fig.png")
plt.show()
```

Figure 1: Jupyter notebook python code for the setup-fee function.

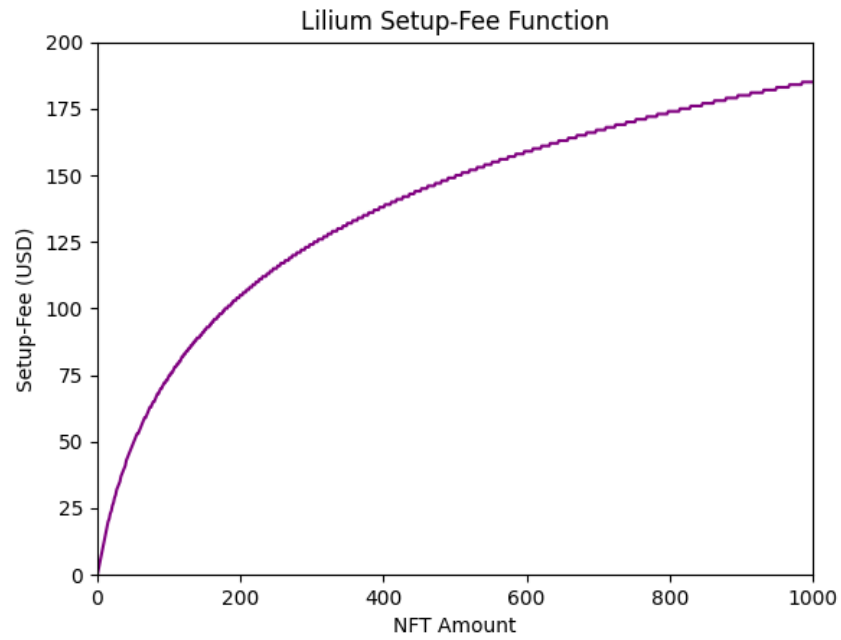


Figure 2: Plot of the setup-fee function: $\phi(x) = \lfloor 54.10 \ln(0.03x + 1) \rfloor$, $x \geq 0$